N72 33913

NASA CR-112174

A THEORETICAL ANALYSIS OF THE FREE VIBRATIONS OF RING- AND/OR STRINGER-STIFFENED ELLIPTICAL CYLINDERS WITH ARBITRARY END CONDITIONS, VOLUME II - USERS MANUAL FOR COMPUTER PROGRAM

By Donald E. Boyd, C. K. P. Rao and Robert L. Brugh

Prepared under Grant No. NGR-37-002-089 at
OKLAHOMA STATE UNIVERSITY
Stillwater, Oklahoma

for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

TABLE OF CONTENTS

ABSTRACT

An analysis was made to determine the natural frequencies and mode shapes of ring- and/or stringer-stiffened noncircular cylinders with arbitrary end conditions. The method of analysis used and the results of the analysis are presented in Volume I of this report (Reference 1). Volume II contains the computer program and the user instructions for the program. Sample input and output is presented in the appendices.

A THEORETICAL ANALYSIS OF THE FREE VIBRATIONS OF RING- AND/OR STRINGER-STIFFENED ELLIPTICAL CYLINDERS WITH ARBITRARY END CONDITIONS, VOLUME II - USERS MANUAL FOR COMPUTER PROGRAM

By Donald E. Boyd, C. K. P. Rao and
Robert L. Brugh

INTRODUCTION

The free vibration characteristics of ring- and/or stringer-stiffened circular and noncircular cylindrical shells are of interest to designers of flight and marine structures. Frequently, fuselages of flight structures and hulls of submarines have noncircular cross-sections due either to special internal storage requirements or to imperfections occurring during manufacture. The method of analysis developed in Volume I of this report (Reference 1) is capable of evaluating the free-vibrational characteristics of ring- and/or stringer-stiffened "singly" symmetric noncircular cylinders with arbitrary end conditions. In this analysis, the stiffeners are treated as discrete elements. The stiffeners may be arbitrarily located and all stiffeners need not possess the same geometric and material properties; however, the stiffeners are assumed to be uniform along their axes. The analysis considers the extension and flexure of the shell and extension, torsion, and flexure

about both cross-section axes of the stiffeners. The stringers may have nonsymmetric cross-sections but the rings are assumed to have "singly" symmetric cross-sections. The rotary inertia of the shell is neglected.

Based on this method of analysis, a computer program was developed. Using this program, a comparative study was made using known solutions for circular and noncircular, unstiffened and stiffened cylinders with various end conditions. Results of this study are presented in Reference 1. The limitations of the program and instructions for using the program are discussed in the following paragraphs of this report.

PROGRAM LIMITATIONS

A flow chart of the main program is given in Appendix A. The main program and some of its subroutines, as listed in Appendix B of this report, are written in single precision for use on the CDC Model 6600 Computer. The mass and stiffness matrices for the entire shell structure are generated in the program. This program uses the subroutine EIGENP (2) to determine the eigenvalues and eigenvectors of the problem. A dictionary of the variables used in the main program is presented in Appendix C.

The computer program has the following limitations:

A. Shell

1. Constant thickness
2. Isotropic material properties

B. Stringers

1. Maximum number; 16
2. Maximum kinds; 1
3. Uniform along length

C. Rings

1. Maximum number; 11
2. Maximum kinds; 2
3. Uniform around circumference
4. "Singly" symmetric about z-axis

D. Number of terms

The maximum number of terms in general must satisfy the following equation.

$$3 \text{ (n terms used) (m terms used)} \leq 90$$

For a specific case, refer to the equations given in the computer program for determining the value of MN3. MN3 must be less than or equal to 90.

The limitations on the program may be made less restrictive by increasing the appropriate dimensions in the dimension statement of the main program.

USER INSTRUCTIONS

In addition to the main program and subroutines listed, the user must supply three function subroutines. The first, FUNCTION RSHL(T), defines the radius of curvature [R] of the shell as a function of the θ coordinate. FUNCTION RRRT(T) defines the first derivative with respect to θ of the reciprocal of the radius $\left[\left(\frac{1}{R}\right)_{,\theta}\right]$. The third, FUNCTION RSHLT(T), defines the first derivative with respect to θ of the radius $[R_{,\theta}]$. As an example, page 79(a) presents the subroutines written for an elliptical cylinder having a specific major (A) and minor (B) axis.

The input data for the program is prepared according to Appendix D. The input data is divided into the following four categories: (1) general data; (2) shell data; (3) stringer data; and (4) ring data. The general and shell data are required for all computer runs. The program in its current state will solve problems with the following boundary conditions: free-free, clamped-free, freely supported, and clamped-clamped. The input variables are defined at the beginning of the program listing.

The other two categories (stringer and ring data) are needed only when the shell structure is stiffened by rings and/or stringers. A set of stringer and/or ring data will be required for each kind of ring and/or stringer used to stiffen the shell.

A computer output for an example problem is presented in Appendix E. The example problem has both stringer and ring stiffening. It should be noted that all input data is given on the printout. The first page gives

the general information and shell data. The second and third pages give the stringer and ring data, respectively. The stringer and ring data pages will appear in the printout only when the stiffening is used in the problem. Other printout options may be selected such that the stiffness matrix, the mass matrix and the eigenvectors may be printed out.

An input listing for this same problem is presented in Appendix F. Cards one through four give the general information. The shell data is on card five. Cards six through thirteen and fourteen through nineteen give the stringer and ring data, respectively. The twentieth card is the first card of the general information of the second problem. The integer "one" (1) punched in column 80 of this card indicates it is the last card of the data set. It should be noted that there is no limit to the number of problems which can be solved in each run.

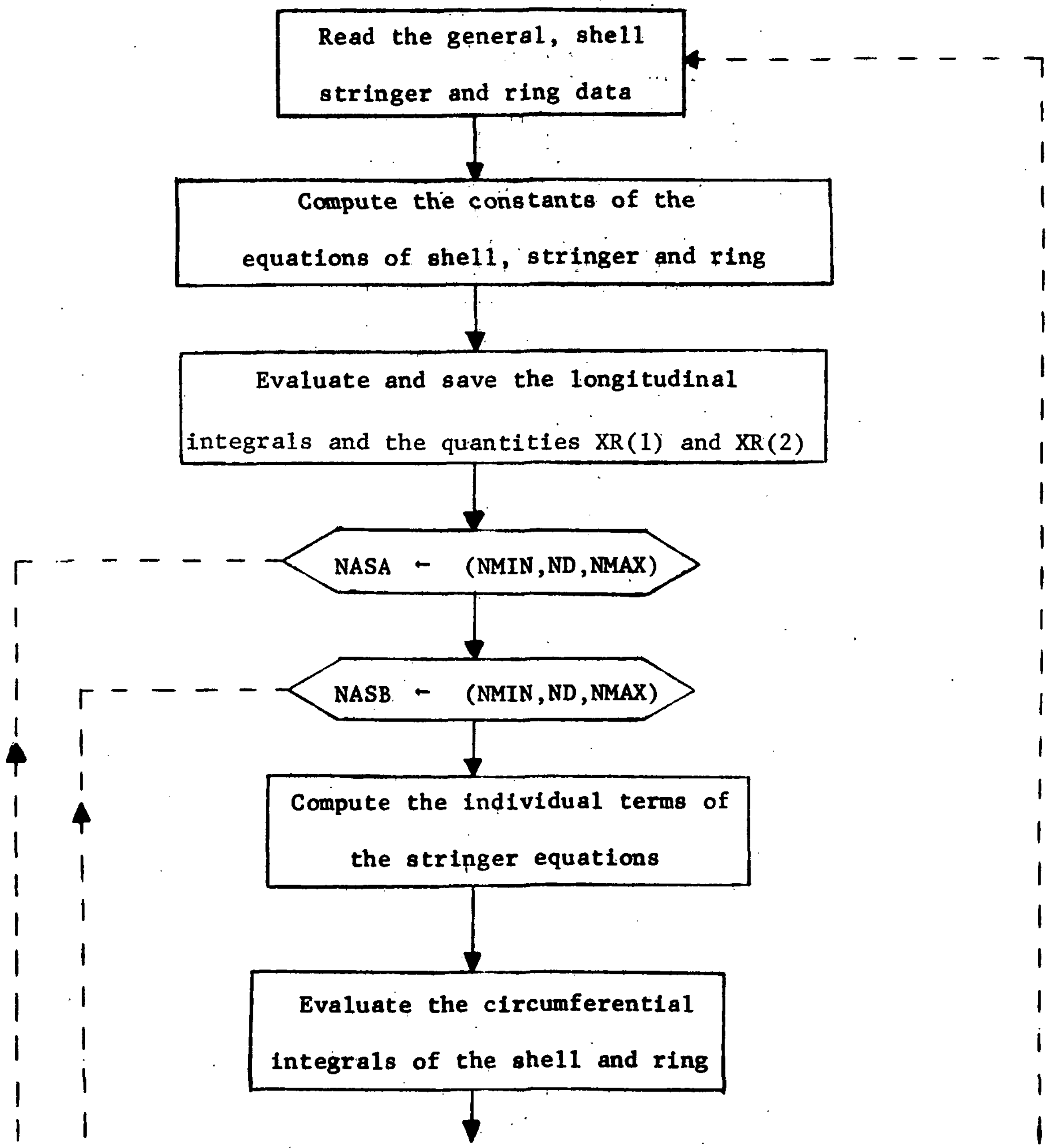
REFERENCES

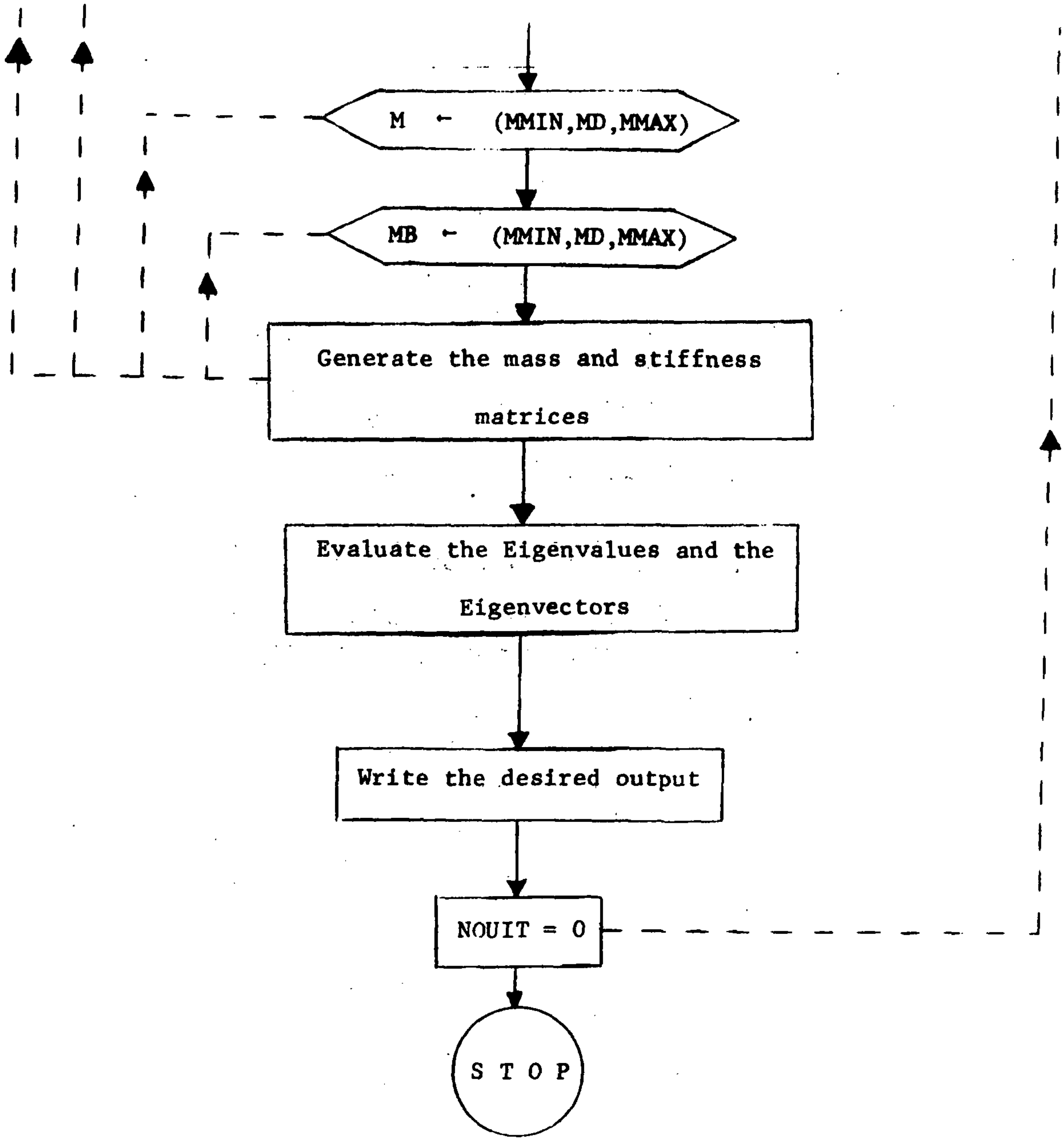
1. Boyd, Donald E., and C. K. P. Rao. "A Theoretical Analysis of the Free Vibrations of Ring- and/or Stringer-Stiffened Elliptical Cylinders With Arbitrary End Conditions, Volume I - Analytical Derivation and Applications." NASA CR-2151, (1972).

2. Grad, J., and M. A. Brebner. "Eigenvalues and Eigenvectors of a Real General Matrix." Communication of the ACM, Vol. 11, No. 12, (December, 1968).

APPENDIX A

FLOW CHART OF THE MAIN PROGRAM




M ← (MMIN,MD,MMAX)
MB ← (MMIN,MD,MMAX)
Generate the mass and stiffness matrices
Evaluate the Eigenvalues and the Eigenvectors
Write the desired output
NOUIT = 0
S T O P

```
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                  +
C     +    REFERENCE                                                     +
C     +                                                                  +
C     +    'FREE VIBRATIONAL ANALYSIS OF STIFFENED OR UNSTIFFENED        +
C     +     CIRCULAR OR NONCIRCULAR CYLINDERS WITH ARBITRARY END         +
C     +                          CONDITIONS'                             +
C     +                                                                  +
C     +    LANGUAGE USED            FORTRAN IV                           +
C     +    DIGITAL MACHINE          IBM 360/65                           +
C     +    PROGRAMMER               C. K. PANDURANGA RAO                 +
C     +                             GRADUATE RESEARCH ASSOCIATE          +
C     +                             SCHOOL OF MECHANICAL AND             +
C     +                               AEROSPACE ENGINEERING              +
C     +                             OKLAHOMA STATE UNIVERSITY            +
C     +                             STILLWATER, OKLAHOMA 74074           +
C     +    DATE OF COMPLETION       AUGUST 30, 1971                      +
C     +                                                                  +
C     +    DESCRIPTION OF THE PROGRAM                                    +
C     +                                                                  +
C     + THIS PROGRAM COMPUTES BOTH THE SYMMETRIC AND ANTISYMMETRIC       +
C     + FREQUENCIES AND THE CORRESPONDING EIGENVECTORS OF STIFFENED OR   +
C     + UNSTIFFENED CIRCULAR OR NONCIRCULAR CYLINDERS WITH ARBITRARY END +
C     + CONDITIONS.  THE RAYLEIGH-RITZ METHOD IS USED TO GENERATE THE    +
C     + STIFFNESS AND MASS MATRICES.                                     +
C     +                                                                  +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                  +
C     +    DESCRIPTION OF THE PARAMETERS                                 +
C     +                                                                  +
C     + INPUT PARAMETERS                                                 +
C     +                                                                  +
C     + BCR          NAME OF THE BOUNDARY CONDITION                      +
C     + NQUIT        1 IN THE  80 TH COLUMN OF A BLANK CARD AT THE END   +
C     +              OF THE DATA SETS TO SIGNIFY THE END OF DATA SETS    +
C     + NG           ORDER OF THE GAUSSIAN QUADRATURE.  NG HAS TO BE     +
C     +              ANY ONE OF THE FOLLOWING NUMBERS  3,4,5,6,7,8,9,10, +
C     +              16, AND 32.                                         +
C     + KG           NUMBER OF CIRCUMFERENTIAL INTERVALS INTO WHICH THE  +
C     +              LIMITS OF INTEGRATION ARE DIVIDED                   +
C     + LL           TOTAL NUMBER OF STRINGERS                           +
C     + NL           NUMBER OF KINDS OF STRINGERS                        +
C     + KK           TOTAL NUMBER OF RINGS                               +
C     + NK           NUMBER OF KINDS OF RINGS                            +
C     + MMIN         STARTING VALUE OF  M  IN THE ASSUMED DISPL SERIES   +
C     + MMAX         FINAL VALUE OF  M  IN THE ASSUMED DISPL SERIES      +
C     + MSA          0 WHEN ONLY EVEN  M  VALUES ARE CONSIDERED          +
C     +              1 WHEN ONLY ODD   M  VALUES ARE CONSIDERED          +
C     +              2 WHEN BOTH EVEN AND ODD VALUES OF M ARE CONSIDERED +
C     + NMIN         STARTING VALUE OF  N  IN THE ASSUMED DISPL SERIES   +
C     + NMAX         FINAL VALUE OF  N  IN THE ASSUMED DISPL SERIES      +
C     + NSA          0 WHEN COMPUTING THE SYMMETRIC MODE SHAPES WITH     +
C     +                RESPECT TO THE VERTICAL AXIS OF THE CROSS-SECTION +
C     +              1 WHEN COMPUTING THE ANTISYMMETRIC MODE SHAPES WITH +
C     +                RESPECT TO THE VERTICAL AXIS OF THE CROSS-SECTION +
C     + NEO          0 WHEN ONLY EVEN  N  VALUES ARE CONSIDERED          +
C     +              1 WHEN ONLY ODD   N  VALUES ARE CONSIDERED          +
C     +              2 WHEN BOTH EVEN AND ODD VALUES OF N ARE CONSIDERED +
C     + IR           0 WHEN THE CROSS-SECTION OF THE SHELL IS CIRCULAR   +
C     +              1 WHEN THE CROSS-SECTION OF THE SHELL IS            +
C     +                NONCIRCULAR                                       +
C     + NKR          NUMBER OF THE KINDS OF RINGS WHICH HAVE DIFFERENT   +
```

```
C    +                    CENTROIDS                                                +
C    + NWK                0 WHEN THE STIFFNESS MATRIX IS NOT TO BE PRINTED         +
C    +                    1 WHEN THE STIFFNESS MATRIX IS TO BE PRINTED             +
C    +                    2 WHEN THE STIFFNESS MATRIX IS TO BE PRINTED ANC         +
C    +                      PUNCHED OUT ON THE CARDS                               +
C    + NWM                0 WHEN THE MASS MATRIX IS NOT TO BE PRINTED              +
C    +                    1 WHEN THE MASS MATRIX IS TO BE PRINTED                  +
C    +                    2 WHEN THE MASS MATRIX IS TO BE PRINTED AND              +
C    +                      PUNCHED OUT ON THE CARDS                               +
C    + NWEV               0 WHEN THE EIGENVECTOR MATRIX IS NOT TO BE PRINTED       +
C    +                    1 WHEN THE EIGENVECTOR MATRIX IS TO BE PRINTED           +
C    +                    2 WHEN THE EIGENVECTOR MATRIX IS TO BE PRINTED AND       +
C    +                      PUNCHED OUT ON THE CARDS                               +
C    + TITLE1             THE TITLE OF THE PROBLEM                                 +
C    + TITLE2             THE TITLE OF THE PROBLEM (CONTINUED)                     +
C    + PC                 THE MASS DENSITY OF THE SHELL                            +
C    + EC                 THE YOUNG'S MODULUS OF THE SHELL                         +
C    + XNU                THE POISSON'S RATIO OF THE SHELL                         +
C    + H                  THE THICKNESS OF THE SHELL                               +
C    + AA                 LONGITUDINAL LENGTH OF THE SHELL                         +
C    + NNL(L)             NUMBER OF STRINGERS WHICH HAVE THE L TH SET OF           +
C    +                    PROPERTIES                                               +
C    + T(L,I)             LIST OF THETA VALUES (IN DEGREES) AT WHICH THE L TH      +
C    +                    SET OF STRINGERS ARE LOCATED                             +
C    + PS(L)              THE MASS DENSITY OF THE  L TH  SET OF STRINGERS          +
C    + ES(L)              THE YOUNG'S MODULUS OF THE  L TH SET OF STRINGERS        +
C    + AS(L)              CROSS-SECTIONAL AREA OF THE  L TH SET OF STRINGERS       +
C    + Z1S(L)             THE Z-DISTANCE OF THE SHEAR CENTER OF THE  L TH SET      +
C    +                    OF STRINGERS FROM THE SHELL'S MIDDLE SURFACE             +
C    + Z2S(L)             THE Z-DISTANCE OF THE CENTROID OF THE  L TH  SET OF      +
C    +                    STRINGERS FROM THEIR SHEAR CENTER                        +
C    + Y1S(L)             THE Y-DISTANCE OF THE SHEAR CENTER OF THE  L TH SET      +
C    +                    OF STRINGERS FROM THE Z-AXIS PASSING THROUGH THEIR       +
C    +                    POINTS OF ATTACHMENT                                     +
C    + Y2S(L)             THE Y-DISTANCE OF THE CENTROID OF THE  L TH SET OF       +
C    +                    STRINGERS FROM THEIR SHEAR CENTER                        +
C    + ZIS(L)             THE MOMENT OF INERTIA OF THE CROSS-SECTION OF THE        +
C    +                    L TH SET OF STRINGERS ABOUT THE Z-AXIS PASSING           +
C    +                    THROUGH THEIR CENTROID                                   +
C    + YIS(L)             THE MOMENT OF INERTIA OF THE CROSS-SECTION OF THE        +
C    +                    L TH SET OF STRINGERS ABOUT THE Y-AXIS PASSING           +
C    +                    THROUGH THEIR CENTROID                                   +
C    + YZIS(L)            THE PRODUCT INERTIA OF THE CROSS-SECTION OF THE          +
C    +                    L TH SET OF STRINGERS ABOUT Y- AND Z-AXES PASSING        +
C    +                    THROUGH THEIR CENTROID                                   +
C    + GJS(L)             THE TORSIONAL STIFFNESS OF THE  L TH SET OF              +
C    +                    STRINGERS                                                +
C    + NNK(K)             NUMBER OF RINGS WHICH HAVE THE K TH SET OF RING          +
C    +                    PROPERTIES                                               +
C    + RX(K,I)            LIST OF X-POSITIONS OF THE RINGS WITH  K TH  SET OF      +
C    +                    RING PROPERTIES                                          +
C    + PR(K)              THE MASS DENSITY OF THE  K TH SET OF RINGS               +
C    + ER(K)              THE YOUNG'S MODULUS OF THE  K TH SET OF RINGS            +
C    + AR(K)              THE CROSS-SECTIONAL AREA OF THE  K TH SET OF RINGS       +
C    + E1R(K)             THE Z-DISTANCE OF THE SHEAR CENTER OF THE  K TH SET      +
C    +                    OF RINGS FROM THE MIDDLE SURFACE OF THE SHELL            +
C    + E2R(K)             THE Z-DISTANCE OF THE CENTROID OF THE  K TH  SET OF      +
C    +                    RINGS FROM THEIR SHEAR CENTER                            +
C    + ZIR(K)             THE MOMENT OF INERTIA OF THE CROSS-SECTION OF THE        +
C    +                    K TH SET OF RINGS ABOUT THE Z-AXIS PASSING THROUGH       +
C    +                    THEIR CENTROID                                           +
C    + XIR(K)             THE MOMENT OF INERTIA OF THE CROSS-SECTION OF THE        +
```

```
C     +              K TH SET OF RINGS ABOUT THE X-AXIS PASSING THROUGH  +
C     +              THEIR CENTROID                                       +
C     + GJR(K)       THE TORSIONAL STIFFNESS OF THE  K TH  SET OF RINGS   +
C     +                                                                   +
C     + THE REMAINING PARAMETERS OF THIS PROGRAM ARE DESCRIBED IN THE     +
C     + DICTIONARY OF VARIABLES                                           +
C     +                                                                   +
C     + SUBROUTINES REQUIRED                                              +
C     +    INTGRL                                                         +
C     +    XX                                                             +
C     +    GAUSS                                                          +
C     +    SHELL1                                                         +
C     +    SHELL2                                                         +
C     +    RING1                                                          +
C     +    RING2                                                          +
C     +    RING3                                                          +
C     +    RING4                                                          +
C     +    RING5                                                          +
C     +    RING6                                                          +
C     +    EIGEN                                                          +
C     +    JACOBI                                                         +
C     +    MATMUL                                                         +
C     +                                                                   +
C     + FUNCTION SUBROUTINES REQUIRED                                     +
C     +    RSHL                                                           +
C     +    RRRT                                                           +
C     +    RSHLT                                                          +
C     +                                                                   +
C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      INTEGER NBC,BC(2,4),BCK(2),TITLE1(7),TITLE2(7)
      DIMENSION T(1,16),PS(1),ES(1),AS(1),Z1S(1),Z2S(1),Y1S(1),Y2S(1),
     1ZIS(1),YIS(1),YZIS(1),GJS(1),RX(2,11),PR(2),ER(2),AR(2),E1R(2),
     2E2R(2),ZIR(2),XIR(2),GJR(2),X(5,55),XXX(2,2,55),C(8),
     3                      SUM(18),NNL(1),NNK(2),ST(75),TS(1,42),SS(1,30)
     4,NNR(2),NR(2,2),CR(2,40),RI(2,54),RCG(2)
      DIMENSION AK(90,90),AM(90,90),VECR(90,90),EVR(90),LC(90)
      DIMENSION XXXX(8100),Y(8100),Z(8100),MC(90),EVI(90),INDIC(90)
      COMMON DR(9),R(9),DRV(5),RV(5),R1(8),RR1(8),R2(10),RR2(10),R3(2),
     1RR3(2),R4(5),RR4(5),R5(18),RR5(18),R6(11),RR6(11),PI,XK,AA,XI(5),
     2XR(2),E1RK,E2RK,N,NB,NBC,K,KB,NSA
      DATA BC/10HCLAMPED-FR,10HEE        ,10HFREELY SUP,10HPORTED    ,10
     1HCLAMPED CL,10HAMPED     ,10HFREE-FREE ,10H          /
C
C     EQUIVALENCE THE STIFFNESS AND EIGENVECTOR MATRICES
C
      EQUIVALENCE(AM(1,1),XXXX(1))
      EQUIVALENCE(AK(1,1),Y(1))
      EQUIVALENCE(INDIC(1),LC(1))
      EQUIVALENCE(VECR(1,1),Z(1),X(1,1)),(VECR(1,11),XXX(1,1,1))
      EXTERNAL SHELL1
      EXTERNAL SHELL2
      EXTERNAL RING1
      EXTERNAL RING2
      EXTERNAL RING3
      EXTERNAL RING4
      EXTERNAL RING5
      EXTERNAL RING6
C
C     KRRR SHOULD BE EQUAL TO THE FIRST DIMENSION OF  AK,AM,VECR
C     MATRICIES
C
      KRRR=90
```

```
10000 ZERO=0.0
      NEXIT=0
C
C     READ THE NAME OF THE BOUNDARY CONDITION
C
      READ(5,8)BCR,NQUIT
    8 FORMAT(2A10,59X,I1)
      IF(NQUIT .NE. 0) CALL EXIT
 2008 WRITE(6,1001)
 1001 FORMAT(1H1,6X,67(1H*),//,6X,65HFREE VIBRATIONAL ANALYSIS OF STIFFE
     1NED OR UNSTIFFENED CIRCULAR OR,/,13X,51HNONCIRCULAR CYLINDERS WITH
     2 ARBITRARY END CONDITIONS,//,6X,67(1H*),////)
C
C     IDENTIFICATION OF THE BOUNDARY CONDITION AND ASSIGNING A CODE
C     NUMBER NBC AS FOLLOWS
C
C     NBC  =  1   FOR CLAMPED-FREE
C     NBC  =  2   FOR FREELY SUPPORTED
C     NBC  =  3   FOR CLAMPED CLAMPED
C     NBC  =  4   FOR FREE-FREE
C
      DO 2 J=1,4
      DO 3 I=1,2
      IF(BCR(I) .EQ. BC(I,J)) GO TO 3
      GO TO 2
    3 CONTINUE
      NBC=J
      GO TO 4
    2 CONTINUE
      WRITE(6,2003) BCR
 2003 FORMAT(//,1X,19H***** ERROR *****  ,27HBOUNDARY CONDITION READ IS
     1,2A10,/,20X,46HTHE BOUNDARY CONDITION MAY NOT BE WORDED RIGHT,/20
     2,37HOR THIS BOUNDARY CONDITION MAY NOT BE,/,20X,25HAVAILABLE IN TH
     3IS PROGRAM  )
      NEXIT=1
C
C     READ AND WRITE THE GENERAL INFORMATION
C
    4 READ(5,60)NG,KG,LL,NL,KK,NK,MMIN,MMAX,MSA,NMIN,NMAX,NSA,NEO,IR,
     1 NWK,NWM,NWEV
   60 FORMAT(20I4)
      WRITE(6,10015) NG,KG,LL,NL,KK,NK,MMIN,MMAX,MSA,NMIN,NMAX,NSA,NEO,
     1IR,NWK,NWM,NWEV
10015 FORMAT(26X,25HGENERAL INPUT INFORMATION,/,26X,25(1H-),/,8X,
     16HNG    =,I4,2X,6HKG    =,I4,2X,6HLL    =,I4,2X,5HNL   =,I4,3X,
     26HKK    =,I4,/,8X,6HNK    =,I4,2X,6HMMIN =,I4,2X,6HMMAX =,I4,2X,
     35HMSA =,I4,3X,6HNMIN =,I4,/,8X,6HNMAX =,I4,2X,6HNSA   =,I4,2X,
     46HNEO   =,I4,2X,5HIR   =,I4,3X,6HNWK   =,I4,/,8X,6HNWM   =,I4,2X,
     56HNWEV =,I4,////)
      PI=3.141592653589793
      PI2=PI+PI
  174 IF(NL .GT. LL) GO TO 176
      GO TO 177
  176 WRITE(6,178) NL,LL
  178 FORMAT(//,1X,19H***** ERROR *****  ,5HNL = ,I4,5X,5HLL = ,I4,/,
     120X,28HNL CANNOT BE GREATER THAN LL    )
      NEXIT=1
  177 IF(NK .GT. KK) GO TO 179
      GO TO 180
  179 WRITE(6,181)NK,KK
  181 FORMAT(//,1X,19H***** ERROR *****  ,5HNK = ,I4,5X,5HKK = ,I4,/,
     120X,28HNK CANNOT BE GREATER THAN KK     )
```

```
      NEXIT=1
C
C     COMPUTE THE CRDER OF THE MASS AND STIFFNESS MATRICES
C
  180 IF(NEXIT .GT. 0) GO TO 10000
      MD=1
      IF(MSA .LT. 2) MD=2
      MS=(MMAX-MMIN)/MD+1
      ND=1
      IF(NEO .LT. 2) ND=2
      NS=(NMAX-NMIN)/ND+1
      MN=MS*NS
      IO=0
      IF(NBC .EQ. 4 .AND. MSA .NE. 1) IO=NS
      IF(NMIN .GT. 0) GO TO 2045
      IF(NSA) 2046,2046,2047
 2046 MN3=3*MN-IO-MS
      GO TO 2048
 2047 MN3=3*MN-IO-2*MS
      GO TO 2048
 2045 MN3=3*MN-IO
C
C      ZERO OUT THE UPPER TRIANGULAR MATRIX OF MASS AND STIFFNESS
C     MATRICIES
C
 2048 DO 2C04 I=1,MN3
      CO 2004 J=1,MN3
      AK(I,J)=0.0
 2CC4 AM(I,J)=0.0
C
C     READ ANC WRITE THE SHELL DATA
C
      REAC(5,1009) TITLE1,TITLE2
 1009 FORMAT(7A10,/,7A10)
      WRITE(6,1003)TITLE1,TITLE2
 1003 FORMAT(29X,19HS H E L L   D A T A,/,29X,19(1H-),////,5X,7A10,//,5X
     1,7A10,////)
      READ(5,65)PC,EC,XNU,H,AA
   65 FCRMAT(5E15.8)
      WRITE(6,1002)PC,EC,XNU,H,AA,BCR
 1002 FORMAT(10X,12HMASS DENSITY,10X,2H= ,E15.8,18H LB SEC.**2/IN.**4//,
     110X,24HMODULUS OF ELASTICITY = ,E15.8,10H LB/IN.**2//,10X,15HPOISS
     2ON'S RATIO,7X,2H= ,E15.8,//,1CX,9HTHICKNESS,13X,2H= ,E15.8,7H INCH
     3ES,//,10X,6HLENGTH,16X,2H= ,E15.8,7H INCHES,//,10X,14HEND CONDITIO
     4NS,8X,3H=  ,2A10)
      PC=PC*H*2.0
      IF(LL .EQ. 0) GO TO 85
C
C     READ AND WRITE THE STRINGER DATA
C
      WRITE(6,1004)LL,NL
 1004 FORMAT(1H1,26X,25HS T R I N G E R   D A T A,/,27X,25(1H-),//,17X,
     143H(THE UNITS ARE SAME AS THOSE OF SHELL DATA),//,23X,28HTOTAL NUM
     2BER OF STRINGERS = ,I4,/,15X,41HNUMBER OF DIFFERENT KINDS OF STRIN
     3GERS = ,I4,/,5X,67(1H=))
      IZ1=0
      IZ2=C
      IY1=0
      IY2=0
      DO 66 L=1,NL
      REAC(5,60)NNL(L)
      NNNL=NNL(L)
```

```
      READ(5,65)(T(L,I),I=1,NNNL)
      READ(5,65)PS(L),ES(L),AS(L),Z1S(L),Z2S(L),Y1S(L),Y2S(L),ZIS(L),
     1YIS(L),YZIS(L),GJS(L)
      IF(Z1S(L) .NE. 0.0   ) IZ1=1
      IF(Z2S(L) .NE. 0.0   ) IZ2=1
      IF(Y1S(L) .NE. 0.0   ) IY1=1
      IF(Y2S(L) .NE. 0.0   ) IY2=1
      WRITE(6,1005)NNL(L),PS(L),ES(L),AS(L),Z1S(L),Y1S(L),Z2S(L),Y2S(L),
     1ZIS(L),YIS(L),YZIS(L),GJS(L)
 1005 FORMAT(//,15X,I4,41H STRINGERS WITH THE FOLLOWING PROPERTIES ,//,
     15X,18HMASS DENSITY     = ,E15.8,2X,17HMOD. OF ELAS.  = ,E15.8,/,5X,
     24HAREA,12X,2H= ,E15.8,2X,17HSHEAR CTR. (Z1)= ,E15.8,/,5X,18HSHEAR
     2CTR. (Y1) = ,E15.8,2X,17HCENTROID (Z2)  = ,E15.8,/,5X,18HCENTROID
     4 (Y2)   = ,E15.8,2X,17HINERTIA (IZZ)   = ,E15.8,/,5X,18HINERTIA  (IY
     5Y)   = ,E15.8,2X,17HPROD. INER.(IYZ)= ,E15.8,/,20X,22HTORSIONAL STIF
     6FNESS = ,E15.8,//,5X,43HLOCATED AT FOLLOWING THETA VALUES (DEGREES
     7),/)
      WRITE(6,1006)(T(L,I),I=1,NNNL)
 1006 FORMAT(4X,E15.8,1X,E15.8,1X,E15.8,1X,E15.8)
      DO 2000 I=1,NNNL
      T(L,I)=T(L,I)*PI/0.18E+03
 2000 CONTINUE
      WRITE(6,1010)
 1010 FORMAT(/,5X,67(1H=))
C
C     COMPUTE THE MOMENT OF INERTIAS WITH RESPECT TO AXES PASSING
C     THROUGH THE SHEAR CENTER OF STRINGERS
C
      IF(Z2S(L) .EQ. 0.0   ) GO TO 182
      ZIS(L)=ZIS(L)+AS(L)*Y2S(L)*Y2S(L)
  182 IF(Y2S(L) .EQ. 0.0   ) GO TO 66
      YIS(L)=YIS(L)+AS(L)*Z2S(L)*Z2S(L)
      YZIS(L)=YZIS(L)+AS(L)*Y2S(L)*Z2S(L)
   66 CONTINUE
C
C     READ AND WRITE THE RING DATA
C
   85 IF(KK .EQ. 0) GO TO 86
      WRITE(6,1007)KK,NK
 1007 FORMAT( 1H1,30X,17HR I N G   D A T A,/,31X,17(1H-),//,17X,43H(THE U
     1NITS ARE SAME AS THOSE OF SHELL DATA),//,24X,24HTOTAL NUMBER OF RI
     2NGS = ,I4,/,17X,37HNUMBER OF DIFFERENT KINDS OF RINGS = ,I4,/,5X,
     367(1H=))
      IE1=0
      IE2=0
      NKR=1
      DO 75 K=1,NK
      READ(5,60)NNK(K)
      NNNK=NNK(K)
      READ(5,65)(RX(K,I),I=1,NNNK)
      READ(5,65)PR(K),ER(K),AR(K),E1R(K),E2R(K),ZIR(K),XIR(K),GJR(K)
      IF(E1R(K) .NE. 0.0   ) IE1=1
      IF(E2R(K) .NE. 0.0   ) IE2=1
      CG=E1R(K)+E2R(K)
      IF(K .EQ. 1) RCG(1)=E1R(1)+E2R(1)
      DO 10008 I=1,NKR
      IF(RCG(I) .EQ. CG) GO TO 10009
10008 CONTINUE
      NKR=NKR+1
      RCG(NKR)=CG
10009 CONTINUE
      WRITE(6,1008)NNK(K),PR(K),ER(K),AR(K),E1R(K),E2R(K),ZIR(K),XIR(K),
```

```
     1GJR(K)
 10C8 FORMAT(//,17X,I4,36H RINGS WITH THE FOLLCWING PROPERTIES,//,5X,18H
     1MASS DENSITY     = ,E15.8,2X,17HMOD. OF ELAST. = ,E15.8/,5X,4HAREA,
     212X,2H= ,E15.8,2X,17HSHEAR CTR. (E1)= ,E15.8/,5X,18HCENTROID (E2)
     3  = ,E15.8,2X,17HINERTIA (IZZ)  = ,E15.8/,5X,18HINERTIA (IXX)  =
     3,E15.8,2X,17HTORS. STIF.(GJ)= ,E15.8,//,5X,38HLOCATED AT FOLLOWING
     4 X VALUES (INCHES),/)
      WRITE(6,1006)(RX(K,I),I=1,NNNK)
      WRITE(6,1010)
      IF(E2R(K) .EQ. 0.0  ) GO TO 75
      XIR(K)=XIR(K)+AR(K)*E2R(K)*E2R(K)
   75 CCNTINUE
C
C      CENTROIDAL INFORMATION OF RINGS
C
      DO 1C010 I=1,NKR
      NNR(I)=0
      CC 10011 K=1,NK
      CG=E1R(K)+E2R(K)
      IF(CG .NE. RCG(I)) GO TO 10011
      NNR(I)=NNR(I)+1
      NR(I,NNR(I))=K
10011 CCNTINUE
10010 CONTINUE
   86 IF(MMIN .GT. 0) GO TO 67
      IF(NBC .LT. 4) GO TO 99
C
C     IF  MMIN  =  0  INCREASE  MMIN AND MMAX BY 1
C
      MMIN=MMIN+1
      MMAX=MMAX+1
   67 IM=0
      NCHNG=0
      IF(NMIN .GT. 0) GO TO 2040
      NMIN=1
      NMAX=NMAX+1
      NCHNG=1
 2040 CCNTINUE
C
C     EVALUATE THE LONGITUDINAL INTEGRALS AND THE X(OUTPUT OF
C     SUBROUTINE XX) VALUES AND STORE THEM
C
      CO 70 M=MMIN,MMAX,MD
      K=M
      IF(MSA .NE. 1 .AND. NBC .EQ. 4) K=M-1
      CO 70 MB=M,MMAX,MD
      KB=MB
      IF(MSA .NE. 1 .AND. NBC .EQ. 4) KB=MB-1
      IM=IM+1
      CALL INTGRL
      CO 80 I=1,5
      X(I,IM)=XI(I)
   8C CONTINUE
      IF(KK .EQ. 0) GO TO 70
      DO 71 I=1,NK
      XXX(1,I,IM)=0.0
      XXX(2,I,IM)=0.0
      NNNK=NNK(I)
      DO 71 KKKK=1,NNNK
      XK=RX(I,KKKK)
      CALL XX
      XXX(1,I,IM)=XXX(1,I,IM)+XR(1)
```

```
      XXX(2,I,IM)=XXX(2,I,IM)+XR(2)
   71 CONTINUE
   70 CONTINUF
C
C     EVALUATE THE CONSTANTS OF THE SHELL EQUATIONS
C
      D=EC*H*H*H/(12.0   *(1.0   -XNU*XNU))
      S5=2.0*D
      S7=S5*XNU
      S3=D*(1.0   -XNU)
      S6=3.0E0*S3
      S8=4.0E0*S3
      S1=12.0EC*S5/(H*H)
      S4=S1*XNU
      S2=S8*3.0E0/(H*H)
      IF(LL .EQ. 0) GO TO 167
C
C     EVALUATE THE CONSTANTS OF THE STRINGER EQUATIONS
C
      DO 35 I=1,NL
      TS(I,1)=PS(I)*AS(I)
      TS(I,2)=PS(I)*Z1S(I)
      TS(I,3)=PS(I)*Y1S(I)
      TS(I,4)=TS(I,2)+TS(I,3)
      TS(I,5)=TS(I,4)+TS(I,4)
      SS(I,1)=ES(I)*AS(I)
      SS(I,2)=ES(I)*Z1S(I)
      SS(I,3)=ES(I)*Y1S(I)
      IF(Z1S(I) .EQ. 0.0   ) GO TO 15
      TS(I,6)=2.0E0*TS(I,1)*Z1S(I)
      TS(I,10)=TS(I,2)*Z1S(I)
      TS(I,7)=TS(I,10)*Z1S(I)
      TS(I,8)=TS(I,10)+TS(I,10)
      TS(I,9)=TS(I,1)*Z1S(I)*Z1S(I)
      TS(I,11)=TS(I,7)+TS(I,7)
      TS(I,12)=TS(I,9)+TS(I,9)
      SS(I,4)=SS(I,1)*Z1S(I)
      SS(I,7)=SS(I,2)*Z1S(I)
      SS(I,6)=SS(I,7)+SS(I,7)
      SS(I,5)=SS(I,7)*Z1S(I)
      SS(I,8)=SS(I,4)*Z1S(I)
   15 IF(Z2S(I) .EQ. 0.0   ) GO TO 20
      TS(I,13)=2.0E0*TS(I,1)*Z2S(I)
      TS(I,14)=TS(I,13)*Z1S(I)
      TS(I,15)=TS(I,14)+TS(I,14)
      SS(I,9)=SS(I,1)*Z2S(I)
      SS(I,10)=SS(I,9)*Z1S(I)*2.0
   20 IF(Y1S(I) .EQ. 0.0   ) GO TO 25
      TS(I,28)=TS(I,1)*Y1S(I)
      TS(I,16)=TS(I,28)+TS(I,28)
      TS(I,17)=TS(I,16)*Z2S(I)
      TS(I,18)=TS(I,28)*Y1S(I)
      TS(I,25)=TS(I,18)+TS(I,18)
      TS(I,26)=TS(I,3)*Y1S(I)
      TS(I,19)=TS(I,26)*Y1S(I)
      TS(I,24)=TS(I,19)+TS(I,19)
      TS(I,20)=TS(I,25)*Z2S(I)
      TS(I,21)=TS(I,16)*Z1S(I)
      TS(I,22)=TS(I,26)+TS(I,26)
      TS(I,23)=TS(I,21)*Z2S(I)
      TS(I,27)=TS(I,23)/2.0
      SS(I,11)=SS(I,1)*Y1S(I)
```

```
      SS(I,12)=SS(I,11)*Z2S(I)
      SS(I,13)=SS(I,11)*Y1S(I)
      SS(I,17)=SS(I,3)*Y1S(I)
      SS(I,14)=SS(I,17)*Y1S(I)
      SS(I,16)=SS(I,11)*Z1S(I)
      SS(I,18)=SS(I,12)*Z1S(I)
      SS(I,19)=SS(I,12)*Y1S(I)
      SS(I,15)=SS(I,19)*2.0
   25 IF(Y2S(I) .EQ. 0.0  ) GO TO 30
      TS(I,42)=TS(I,1)*Y2S(I)
      TS(I,29)=TS(I,42)+TS(I,42)
      TS(I,30)=TS(I,29)*Z1S(I)
      TS(I,31)=TS(I,29)*Y1S(I)
      TS(I,32)=TS(I,31)*Z1S(I)
      TS(I,35)=TS(I,30)*Z1S(I)
      TS(I,39)=TS(I,31)+TS(I,31)
      TS(I,40)=TS(I,35)/2.0
      TS(I,36)=2.0*PS(I)*YZIS(I)
      TS(I,33)=TS(I,36)*Y1S(I)
      TS(I,34)=TS(I,33)*Z1S(I)
      TS(I,37)=TS(I,36)*Z1S(I)
      TS(I,38)=2.0*TS(I,34)
      TS(I,41)=PS(I)*YZIS(I)*Z1S(I)
      SS(I,20)=SS(I,1)*Y2S(I)
      SS(I,21)=SS(I,20)*Z1S(I)
      SS(I,22)=SS(I,20)*2.0E0*Y1S(I)
      SS(I,23)=SS(I,22)*Z1S(I)
      SS(I,24)=SS(I,23)/2.0
      SS(I,25)=SS(I,21)*Z1S(I)
   30 IF(YZIS(I) .EQ. 0.0  ) GO TO 35
      SS(I,28)=ES(I)*YZIS(I)
      SS(I,29)=SS(I,28)*Z1S(I)
      SS(I,30)=SS(I,28)*Y1S(I)
      SS(I,26)=SS(I,30)+SS(I,30)
      SS(I,27)=SS(I,26)*Z1S(I)
   35 CONTINUE
  167 IF(KK .EQ. 0) GO TO 168
C
C     EVALUATE THE CONSTANTS OF THE RING EQUATIONS
C
      DO 125 K=1,NK
      CR(K,1)=2.0E0*ER(K)*ZIR(K)
      CR(K,2)=2.0E0*ER(K)*AR(K)
      CR(K,3)=2.0E0*ER(K)*XIR(K)
      CR(K,21)=2.0E0*GJR(K)
      CR(K,22)=2.0E0*PR(K)*AR(K)
      CR(K,23)=2.0E0*PR(K)*ZIR(K)
      CR(K,24)=2.0E0*PR(K)*XIR(K)
      IF(E1R(K) .EQ. 0.0  ) GO TO 126
      CR(K,4)=CR(K,1)*E1R(K)
      CR(K,9)=CR(K,2)*E1R(K)
      CR(K,5)=CR(K,9)*E1R(K)
      CR(K,6)=CR(K,3)*E1R(K)*E1R(K)
      CR(K,7)=CR(K,9)+CR(K,9)
      CR(K,8)=2.0E0*E1R(K)*CR(K,3)
      CR(K,10)=CR(K,4)*E1R(K)
      CR(K,25)=CR(K,21)*E1R(K)
      CR(K,26)=CR(K,25)*E1R(K)
      CR(K,27)=CR(K,25)+CR(K,25)
      CR(K,31)=CR(K,22)*E1R(K)
      CR(K,30)=CR(K,31)*E1R(K)
      CR(K,28)=CR(K,31)+CR(K,31)
```

```
      CR(K,29)=CR(K,23)*2.0E0*E1R(K)
      CR(K,32)=CR(K,24)*E1R(K)*E1R(K)
      CR(K,33)=CR(K,24)*2.0E0*E1R(K)
      CR(K,34)=CR(K,23)*E1R(K)*E1R(K)
  126 IF(E2R(K) .EQ. 0.0   ) GO TO 127
      CR(K,14)=CR(K,2)*E2R(K)
      CR(K,11)=CR(K,14)+CR(K,14)
      CR(K,16)=CR(K,14)*E1R(K)
      CR(K,17)=CR(K,16)+CR(K,16)
      CR(K,12)=CR(K,17)*E1R(K)
      CR(K,13)=CR(K,17)+CR(K,17)
      CR(K,15)=CR(K,17)+CR(K,16)
      CR(K,36)=CR(K,22)*E2R(K)
      CR(K,35)=CR(K,36)+CR(K,36)
      CR(K,40)=CR(K,35)*E1R(K)
      CR(K,38)=CR(K,40)+CR(K,40)
      CR(K,39)=CR(K,40)+CR(K,38)
      CR(K,37)=CR(K,40)*E1R(K)
  127 IF(IR .EQ. 0) GO TO 125
      CR(K,19)=CR(K,3)*E1R(K)
      CR(K,18)=CR(K,6)+CR(K,6)
      CR(K,20)=CR(K,16)*E1R(K)
  125 CONTINUE
  168 NDC=0
C
C      THE DO LOOP OF   N
C
      DO 90 NASA=NMIN,NMAX,ND
      N=NASA
      IF(NCHNG .NE. 0)N=NASA-1
      NDC=NDC+1
      AN= FLOAT(N)
      AN2=AN*AN
      NEC=0
C
C      THE DO LOOP OF   NB
C
      DO 91 NASB=NMIN,NMAX,ND
      NB=NASB
      IF(NCHNG .NE. 0)NB=NASB-1
      BN= FLOAT(NB)
      NEC=NEC+1
      BN2=BN*BN
      ABN=AN*BN
      ABN2=ABN*ABN
      ABNB=ABN*BN
      ABNA=ABN*AN
      IF(LL .EQ. 0) GO TO 169
C
C      EVALUATE THE CIRCUMFERENTIAL QUANTITIES OF THE STRINGER
C       75 IS THE TOTAL NUMBER OF TERMS IN THE STRINGER ENERGIES
C
      DO 94 IL=1,75
      ST(IL)=0.0
   94 CONTINUE
      DO 95 L=1,NL
      NNN=NNL(L)
      DO 101 KI=1,8
  101 C(KI)=0.0
      DO 96 LI=1,NNN
      TN=AN*T(L,LI)
      TNB=BN*T(L,LI)
```

```
      IF(NSA .EQ. 1) GO TO 97
      CN= COS(TN)
      CNB= COS(TNB)
      SN= SIN(TN)
      SNB= SIN(TNB)
      GO TO 98
   97 CN= SIN(TN)
      CNB= SIN(TNB)
      SN= COS(TN)
      SNB= COS(TNB)
   98 CC=CN*CNB
      CS=CN*SNB
      SSS=SN*SNB
      SC=SN*CNB
      SR=RSHL(T(L,LI))
      SR2=SR*SR
      C(1)=C(1)+CC
      C(3)=C(3)+SSS
      IF(Z1S(L) .EQ. 0.0  ) GO TO 96
      C(2)=C(2)+CS
      C(4)=C(4)+SC
      C(5)=C(5)+SSS/SR
      C(6)=C(6)+SSS/SR2
      C(7)=C(7)+CS/SR
      C(8)=C(8)+SC/SR
   96 CONTINUE
      ST(1)=ST(1)+SS(L,1)*C(1)
      ST(2)=ST(2)+SS(L,2)*C(3)
      ST(3)=ST(3)+SS(L,3)*C(1)
      ST(73)=ST(73)+GJS(L)*ABN*C(6)
      ST(74)=ST(74)+GJS(L)*C(6)
      ST(75)=ST(75)+GJS(L)*BN*C(6)
      ST(29)=ST(29)+TS(L,1)*C(1)
      ST(30)=ST(30)+TS(L,2)*C(3)
      ST(31)=ST(31)+TS(L,1)*C(3)+TS(L,4)*C(6)
      ST(32)=ST(32)+TS(L,5)*BN*C(6)
      ST(33)=ST(33)+TS(L,3)*C(1)
      ST(34)=ST(34)+ABN*TS(L,4)*C(6)
      ST(35)=ST(35)+TS(L,1)*C(1)
      IF(Z1S(L) .EQ. 0.0  ) GO TO 102
      ST(4)=ST(4)-SS(L,4)*C(1)
      ST(5)=ST(5)+SS(L,5)*C(6)+SS(L,6)*C(5)
      ST(6)=ST(6)+BN*(SS(L,5)*C(6)+SS(L,7)*C(5))
      ST(7)=ST(7)+SS(L,8)*C(1)
      ST(8)=ST(8)+SS(L,5)*C(6)*ABN
      ST(36)=ST(36)-TS(L,6)*C(1)
      ST(37)=ST(37)+TS(L,7)*C(6)+TS(L,8)*C(5)
      ST(38)=ST(38)+TS(L,9)*C(6)+TS(L,6)*C(5)
      ST(39)=ST(39)+(TS(L,11)*C(6)+TS(L,8)*C(5))*BN
      ST(40)=ST(40)+(TS(L,12)*C(6)+TS(L,6)*C(5))*BN
      ST(41)=ST(41)+TS(L,9)*C(1)
      ST(42)=ST(42)+ABN*TS(L,7)*C(6)
      ST(43)=ST(43)+ABN*TS(L,9)*C(6)
  102 IF(Z2S(L) .EQ. 0.0  ) GO TO 103
      ST(9)=ST(9)-SS(L,9)*C(1)
      ST(10)=ST(10)+SS(L,10)*C(1)
      ST(44)=ST(44)-TS(L,13)*C(1)
      ST(45)=ST(45)+TS(L,13)*C(5)+TS(L,14)*C(6)
      ST(46)=ST(46)+(TS(L,13)*C(5)+TS(L,15)*C(6))*BN
      ST(47)=ST(47)+ABN*TS(L,14)*C(6)
      ST(48)=ST(48)+TS(L,14)*C(1)
  103 IF(Y1S(L) .EQ. 0.0  ) GO TO 104
```

```
      ST(11)=ST(11)-SS(L,11)*C(2)+SS(L,12)*C(7)
      ST(12)=ST(12)+BN*SS(L,12)*C(7)
      ST(13)=ST(13)+SS(L,13)*C(3)+SS(L,14)*C(6)-SS(L,15)*C(5)
      ST(14)=ST(14)+(SS(L,12)+SS(L,16))*C(4)-(SS(L,17)+SS(L,18))*C(8)
      ST(15)=ST(15)+BN*(SS(L,14)*C(6)-SS(L,19)*C(5))
      ST(16)=ST(16)+ABN*SS(L,14)*C(6)
      ST(17)=ST(17)-(SS(L,17)+SS(L,18))*(BN*C(7)+AN*C(8))
      ST(49)=ST(49)-TS(L,16)*C(2)+TS(L,17)*C(7)
      ST(50)=ST(50)+BN*TS(L,17)*C(7)
      ST(51)=ST(51)+TS(L,18)*C(3)+TS(L,19)*C(6)-TS(L,20)*C(5)
      ST(52)=ST(52)+TS(L,18)*C(6)
      ST(53)=ST(53)+(TS(L,21)+TS(L,17))*C(4)-(TS(L,22)+TS(L,23))*C(8)
      ST(54)=ST(54)+(TS(L,24)*C(6)-TS(L,20)*C(5))*BN
      ST(55)=ST(55)+TS(L,25)*BN*C(6)
      ST(56)=ST(56)-TS(L,16)*C(8)
      ST(57)=ST(57)+ABN*TS(L,19)*C(6)
      ST(58)=ST(58)-(TS(L,26)+TS(L,27))*(BN*C(7)+AN*C(8))
      ST(59)=ST(59)+ABN*TS(L,18)*C(6)
      ST(60)=ST(60)-TS(L,28)*(BN*C(7)+AN*C(8))
  104 IF(Y2S(L) .EQ. 0.0   ) GO TO 105
      ST(18)=ST(18)-SS(L,20)*C(2)-SS(L,21)*C(7)
      ST(19)=ST(19)-BN*SS(L,21)*C(7)
      ST(20)=ST(20)+SS(L,22)*C(3)+SS(L,23)*C(5)
      ST(21)=ST(21)+SS(L,24)*BN*C(5)
      ST(22)=ST(22)+SS(L,21)*C(4)+SS(L,25)*C(8)
      ST(23)=ST(23)+SS(L,25)*(BN*C(7)+AN*C(8))
      ST(61)=ST(61)-TS(L,29)*C(2)-TS(L,30)*C(7)
      ST(62)=ST(62)-BN*TS(L,30)*C(7)
      ST(63)=ST(63)+TS(L,31)*C(3)+(TS(L,32)-TS(L,33))*C(5)-TS(L,34)*C(6)
      ST(64)=ST(64)+TS(L,31)*C(6)
      ST(65)=ST(65)+(TS(L,30)+TS(L,36))*C(4)+(TS(L,35)+TS(L,37))*C(8)
      ST(66)=ST(66)-BN*((-TS(L,32)+TS(L,33))*C(5)+TS(L,38)*C(6))
      ST(67)=ST(67)-TS(L,29)*C(8)
      ST(68)=ST(68)+BN*TS(L,39)*C(6)
      ST(69)=ST(69)+(TS(L,40)+TS(L,41))*(BN*C(7)+AN*C(8))
      ST(70)=ST(70)-ABN*TS(L,34)*C(6)
      ST(71)=ST(71)+ABN*TS(L,31)*C(6)
      ST(72)=ST(72)-TS(L,42)*(BN*C(7)+AN*C(8))
  105 IF(YZIS(L) .EQ. 0.0   ) GO TO 95
      ST(24)=ST(24)-SS(L,26)*C(5)-SS(L,27)*C(6)
      ST(25)=ST(25)+SS(L,28)*C(4)+SS(L,29)*C(8)
      ST(26)=ST(26)-BN*(SS(L,30)*C(5)+SS(L,27)*C(6))
      ST(27)=ST(27)+SS(L,29)*(BN*C(7)+AN*C(8))
      ST(28)=ST(28)-ABN*SS(L,27)*C(6)
   95 CONTINUE
C
C       INTEGRALS  OF  SHELL
C
  169 CALL GAUSS(NG,KG,ZERO ,PI,9,PHI,DR,SUM,R,SHELL1)
      DO 131 IQ=1,9
      IF( ABS(R(IQ)) .LE. 0.1E-08) R(IQ)=0.0
  131 CONTINUE
      IF(IR .EQ. 0) GO TO 130
      CALL GAUSS(NG,KG,ZERO ,PI,5,PHI,DRV,SUM,RV,SHELL2)
      DO 132 IQ=1,5
      IF( ABS(RV(IQ)) .LE. 0.1E-08) RV(IQ)=0.0
  132 CONTINUE
C
C       INTEGRALS  OF  RING
C
  130 IF(KK .EQ. 0) GO TO 133
      DO 146 LA=1,NK
```

```
      DO 146 LB=1,54
  146 RI(LA,LB)=0.0
      DO 136 KKI=1,NKR
      NRKKI1=NR(KKI,1)
      E1RK=E1R(NRKKI1)
      E2RK=E2R(NRKKI1)
      KQ=NNR(KKI)
      CALL GAUSS(NG,KG,ZERO ,PI,8,PHI,R1,SUM,RR1,RING1)
      DO 135 IQ=1,8
      IF( ABS(RR1(IQ)) .EQ. 0.1E-08) RR1(IQ)=0.0
      DO 135 KR=1,KQ
      NKKIKR=NR(KKI,KR)
      RI(NKKIKR,IQ)=RR1(IQ)
  135 CONTINUE
      IF(E1RK .EQ. 0.0  ) GO TO 137
      CALL GAUSS(NG,KG,ZERO ,PI,10,PHI,R2,SUM,RR2,RING2)
      DO 138 IQ=1,10
      IF( ABS(RR2(IQ)) .LE. 0.1E-08) RR2(IQ)=0.0
      DO 138 KR=1,KQ
      NKKIKR=NR(KKI,KR)
      RI(NKKIKR,IQ+8)=RR2(IQ)
  138 CONTINUE
  137 IF(E2RK .EQ. 0.0  ) GO TO 134
      CALL GAUSS(NG,KG,ZERO ,PI,2,PHI,R3,SUM,RR3,RING3)
      IF( ABS(RR3(1)) .LE. 0.1E-08) RR3(1)=0.0
      IF( ABS(RR3(2)) .LE. 0.1E-08) RR3(2)=0.0
      DO 145 KR=1,KQ
      NKKIKR=NR(KKI,KR)
      RI(NKKIKR,19)=RR3(1)
      RI(NKKIKR,20)=RR3(2)
  145 CONTINUE
  134 IF(IR .EQ. 0) GO TO 139
      CALL GAUSS(NG,KG,ZERO ,PI,5,PHI,R4,SUM,RR4,RING4)
      DO 140 IQ=1,5
      IF( ABS(RR4(IQ)) .LE. 0.1E-08) RR4(IQ)=0.0
      DO 140 KR=1,KQ
      NKKIKR=NR(KKI,KR)
      RI(NKKIKR,IQ+20)=RR4(IQ)
  140 CONTINUE
      IF(E1RK .EQ. 0.0  ) GO TO 141
      CALL GAUSS(NG,KG,ZERO ,PI,18,PHI,R5,SUM,RR5,RING5)
      DO 142 IQ=1,18
      IF( ABS(RR5(IQ)) .LE. 0.1E-08) RR5(IQ)=0.0
      DO 142 KR=1,KQ
      NKKIKR=NR(KKI,KR)
      RI(NKKIKR,IQ+25)=RR5(IQ)
  142 CONTINUE
  141 IF(E2RK .EQ. 0.0  ) GO TO 136
      CALL GAUSS(NG,KG,ZERO ,PI,11,PHI,R6,SUM,RR6,RING6)
      DO 143 IQ=1,11
      IF( ABS(RR6(IQ)) .LE. 0.1E-08) RR6(IQ)=0.0
      DO 143 KR=1,KQ
      NKKIKR=NR(KKI,KR)
      RI(NKKIKR,IQ+43)=RR6(IQ)
  143 CONTINUE
  139 CONTINUE
  136 CONTINUE
  133 IBC=0
C
C     THE DO LOOP OF  M
C
      DO 52 M=MMIN,MMAX,MD
      IF(NCHNG  .EQ. 0) GO TO 2030
      IF(NSA) 2031,2031,2032
 2031 I=NDC+IBC*NS-10
      IBC=IBC+1
      IN=I+MN-IBC
      INN=I+MN+MN-MS
      GO TO 2033
```

```
 2032 ITEMP=NDC+IBC*NS-IO
      IBC=IBC+1
      I=ITEMP-IBC
      IN=ITEMP+MN-MS
      INN=IN+MN-IBC
      GO TO 2033
 2030 I=NDC+IBC*NS-IO
      IBC=IBC+1
      IN=I+MN
      INN=IN+MN
 2033 JBC=0
C
C     THE DO LOOP OF MB
C
      DO 93 MB=MMIN,MMAX,MD
      IF(NCHNG .EQ. 0) GO TO 2034
      IF(NSA)2035,2035,2036
 2035 J=NEC+JBC*NS-IO
      JBC=JBC+1
      JN=J+MN-JBC
      JNN=J+MN+MN-MS
      GO TO 2037
 2036 JTEMP=NEC+JBC*NS-IO
      JBC=JBC+1
      J=JTEMP-JBC
      JN=JTEMP+MN-MS
      JNN=JN+MN-JBC
      GO TO 2037
 2034 J=NEC+JBC*NS-IO
      JBC=JBC+1
      JN=J+MN
      JNN=JN+MN
 2037 IM=(IBC-1)*MS-IBC*(IBC-1)/2+JBC
      IF(JBC .GE. IBC) GO TO 120
      IM=(JBC-1)*MS-JBC*(JBC-1)/2+IBC
      X3=X(4,IM)
      X4=X(3,IM)
      GO TO 121
  120 X3=X(3,IM)
      X4=X(4,IM)
  121 X1=X(1,IM)
      X2=X(2,IM)
      X5=X(5,IM)
C
C     COMPUTE THE UPPER DIAGONAL ELEMENTS OF THE MASS AND STIFFNESS
C     MATRICIES
C
C
C       CONTRIBUTIONS OF SHELL
C
      IF(J .LT. I) GO TO 112
      IF(J .LE. 0 .OR. I.LE. 0) GO TO 114
C       SUBMATRIX  A
      AK(I,J)=AK(I,J)+S1*R(1)*X1+(S2*R(2)+S3*R(3))*ABN*X2
C       SUBMATRIX  N
      AM(I,J)=AM(I,J)+PC*R(1)*X2
C       SUBMATRIX  B
  114 AK(IN,JN)=AK(IN,JN)+S1*ABN*R(8)*X5+(S2*R(9)+S6*R(2))*X2
C       SUBMATRIX  Q
      AM(IN,JN)=AM(IN,JN)+PC*R(9)*X5
C       SUBMATRIX  C
      AK(INN,JNN)=AK(INN,JNN)+(S1*R(8)+S5*R(4))*X5+S5*(R(1)*X1+(ABN2-AN2
```

```
     1-BN2)*R(4)*X5)-S7*R(8)*(BN2*X3+AN2*X4)+S8*ABN*R(2)*X2
C       SUBMATRIX  S
      AM(INN,JNN)=AM(INN,JNN)+PC*R(1)*X5
  112 IF(I .LE. 0) GO TO 113
C       SUBMATRIX  D
      AK(I,JN)=AK(I,JN)+S4*BN*R(5)*X3-S2*AN*R(6)*X2
C       SUBMATRIX  E
      AK(I,JNN)=AK(I,JNN)+R(5)*(S4*X3-S5*X1)+S3*ABN*R(7)*X2
C       SUBMATRIX  F
  113 AK(IN,JNN)=AK(IN,JNN)+AN*R(8)*(S1*X5-S7*X4)+S6*BN*R(2)*X2
      IF(IR .EQ. 0) GO TO 111
      IF(J .LT. I) GO TO 115
C       SUBMATRIX  B
      AK(IN,JN)=AK(IN,JN)+S5*RV(1)*X5
C       SUBMATRIX  F
  115 AK(IN,JNN)=AK(IN,JNN)-S7*RV(3)*X4+S5*(BN*RV(1)-(1.0  -BN2)*RV(2))*
     1X5
C       SUBMATRIX  C
      AK(INN,JNN)=AK(INN,JNN)+S5*(ABN*RV(1)+(ABNB-AN)*RV(2)+(ABNA-BN)*RV
     1(4))*X5-S7*(BN*RV(5)*X3+AN*RV(3)*X4)
C
C       CONTRIBUTIONS OF STRINGER
C
  111 IF(LL .EQ. 0) GO TO 106
      IF(J .LT. I) GO TO 116
      IF(J .LE. 0 .OR. I .LE. 0) GO TO 117
C       SUBMATRIX  A
      AK(I,J)=AK(I,J)+ST(1)*X1
C       SUBMATRIX  N
      AM(I,J)=AM(I,J)+ST(29)*X2
C       SUBMATRIX  B
  117 AK(IN,JN)=AK(IN,JN)+ST(2)*X1+ST(74)*X2
C       SUBMATRIX  Q
      AM(IN,JN)=AM(IN,JN)+ST(30)*X2+ST(31)*X5
C       SUBMATRIX  C
      AK(INN,JNN)=AK(INN,JNN)+ST(3)*X1+ST(73)*X2
C       SUBMATRIX  S
      AM(INN,JNN)=AM(INN,JNN)+ST(33)*X2+(ST(34)+ST(35))*X5
C       SUBMATRIX  R
  116 AM(IN,JNN)=AM(IN,JNN)+ST(32)*X5
C       SUBMATRIX  F
      AK(IN,JNN)=AK(IN,JNN)+ST(75)*X2
      IF(IZ1 .EQ. 0) GO TO 107
      IF(J .LT. I) GO TO 118
C       SUBMATRIX  B
      AK(IN,JN)=AK(IN,JN)+ST(5)*X1
C       SUBMATRIX  Q
      AM(IN,JN)=AM(IN,JN)+ST(37)*X2+ST(38)*X5
C       SUBMATRIX  C
      AK(INN,JNN)=AK(INN,JNN)+(ST(7)+ST(8))*X1
C       SUBMATRIX  S
      AM(INN,JNN)=AM(INN,JNN)+(ST(41)+ST(42))*X2+ST(43)*X5
  118 IF(I .LE. 0) GO TO 119
C       SUBMATRIX  E
      AK(I,JNN)=AK(I,JNN)+ST(4)*X1
C       SUBMATRIX  P
      AM(I,JNN)=AM(I,JNN)+ST(36)*X2
C       SUBMATRIX  F
  119 AK(IN,JNN)=AK(IN,JNN)+ST(6)*X1
C       SUBMATRIX  R
      AM(IN,JNN)=AM(IN,JNN)+ST(39)*X2+ST(40)*X5
  107 IF(IZ2 .EQ. 0) GO TO 108
      IF(J .LT. I) GO TO 520
C       SUBMATRIX  Q
      AM(IN,JN)=AM(IN,JN)+ST(45)*X5
C       SUBMATRIX  C
```

```
      AK(INN,JNN)=AK(INN,JNN)+ST(10)*X1
C       SUBMATRIX  S
      AM(INN,JNN)=AM(INN,JNN)+ST(47)*X5+ST(48)*X2
  520 IF(I .LE. 0) GO TO 521
C       SUBMATRIX  E
      AK(I,JNN)=AK(I,JNN)+ST(9)*X1
C       SUBMATRIX  P
      AM(I,JNN)=AM(I,JNN)+ST(44)*X2
C       SUBMATRIX  R
  521 AM(IN,JNN)=AM(IN,JNN)+ST(46)*X5
  108 IF(IY1 .EQ. 0) GO TO 109
      IF(J .LT. I) GO TO 122
C       SUBMATRIX  B
      AK(IN,JN)=AK(IN,JN)+ST(13)*X1
C       SUBMATRIX  Q
      AM(IN,JN)=AM(IN,JN)+ST(51)*X2+ST(52)*X5
C       SUBMATRIX  C
      AK(INN,JNN)=AK(INN,JNN)+(ST(16)+ST(17))*X1
C       SUBMATRIX  S
      AM(INN,JNN)=AM(INN,JNN)+(ST(57)+ST(58))*X2+(ST(59)+ST(60))*X5
  122 IF(I .LE. 0) GO TO 123
C       SUBMATRIX  D
      AK(I,JN)=AK(I,JN)+ST(11)*X1
C       SUBMATRIX  NN
      AM(I,JN)=AM(I,JN)+ST(49)*X2
C       SUBMATRIX  E
      AK(I,JNN)=AK(I,JNN)+ST(12)*X1
C       SUBMATRIX  P
      AM(I,JNN)=AM(I,JNN)+ST(50)*X2
C       SUBMATRIX  F
  123 AK(IN,JNN)=AK(IN,JNN)+(ST(14)+ST(15))*X1
C       SUBMATRIX  R
      AM(IN,JNN)=AM(IN,JNN)+(ST(53)+ST(54))*X2+(ST(55)+ST(56))*X5
  109 IF(IY2 .EQ. 0) GO TO 110
      IF(J .LT. I) GO TO 124
C       SUBMATRIX  B
      AK(IN,JN)=AK(IN,JN)+ST(20)*X1
C       SUBMATRIX  Q
      AM(IN,JN)=AM(IN,JN)+ST(63)*X2+ST(64)*X5
C       SUBMATRIX  C
      AK(INN,JNN)=AK(INN,JNN)+ST(23)*X1
C       SUBMATRIX  S
      AM(INN,JNN)=AM(INN,JNN)+(ST(69)+ST(70))*X2+(ST(71)+ST(72))*X5
  124 IF(I .LE. 0) GO TO 525
C       SUBMATRIX  D
      AK(I,JN)=AK(I,JN)+ST(18)*X1
C       SUBMATRIX  NN
      AM(I,JN)=AM(I,JN)+ST(61)*X2
C       SUBMATRIX  E
      AK(I,JNN)=AK(I,JNN)+ST(19)*X1
C       SUBMATRIX  P
      AM(I,JNN)=AM(I,JNN)+ST(62)*X2
C       SUBMATRIX  F
  525 AK(IN,JNN)=AK(IN,JNN)+(ST(21)+ST(22))*X1
C       SUBMATRIX  R
      AM(IN,JNN)=AM(IN,JNN)+(ST(65)+ST(66))*X2+(ST(67)+ST(68))*X5
  110 IF(IZ1 .EQ. 0 .OR. IY1 .EQ. 0) GO TO 106
      IF(J .LT. I) GO TO 526
C       SUBMATRIX  B
      AK(IN,JN)=AK(IN,JN)+ST(24)*X1
C       SUBMATRIX  C
      AK(INN,JNN)=AK(INN,JNN)+(ST(27)+ST(28))*X1
```

```
C       SUBMATRIX   F
  526 AK(IN,JNN)=AK(IN,JNN)+(ST(25)+ST(26))*X1
C
C       CONTRIBUTIONS  OF  RING
C
  106 IF(KK .EQ. 0) GO TO 93
      DO 144 KR=1,NK
      XX1=XXX(1,KR,IM)
      XX2=XXX(2,KR,IM)
      IF(J .LT. I) GO TO 151
      IF(J .LE. 0 .OR. I .LE. 0) GO TO 153
C       SUBMATRIX   A
      AK(I,J)=AK(I,J)+(CR(KR,1)*ABN2*RI(KR,1)+CR(KR,21)*ABN*RI(KR,5))*
     1XX1
C       SUBMATRIX   N
      AM(I,J)=AM(I,J)+(CR(KR,22)*RI(KR,4)+CR(KR,23)*RI(KR,7)*ABN)*XX1
C       SUBMATRIX   B
  153 AK(IN,JN)=AK(IN,JN)+(CR(KR,2)*RI(KR,3)+CR(KR,3)*RI(KR,1))*ABN*XX2
C       SUBMATRIX   Q
      AM(IN,JN)=AM(IN,JN)+(CR(KR,22)*RI(KR,8)+CR(KR,24)*RI(KR,7))*XX2
C       SUBMATRIX   C
      AK(INN,JNN)=AK(INN,JNN)+(CR(KR,3)*ABN2*RI(KR,1)+CR(KR,2)*RI(KR,3))
     1*XX2+(CR(KR,1)*RI(KR,3)+CR(KR,21)*ABN*RI(KR,7))*XX1
C       SUBMATRIX   S
      AM(INN,JNN)=AM(INN,JNN)+(CR(KR,24)+CR(KR,23))*RI(KR,4)*XX1+(CR(KR,
     124)*ABN*RI(KR,7)+CR(KR,22)*RI(KR,4))*XX2
  151 IF(I .LE. 0) GO TO 152
C       SUBMATRIX   E
      AK(I,JNN)=AK(I,JNN)+(CR(KR,1)*AN2*RI(KR,2)+CR(KR,21)*ABN*RI(KR,6))
     1*XX1
C       SUBMATRIX   F
  152 AK(IN,JNN)=AK(IN,JNN)+(CR(KR,3)*ABNB*RI(KR,1)+CR(KR,2)*AN*RI(KR,3)
     1)*XX2
C       SUBMATRIX   R
      AM(IN,JNN)=AM(IN,JNN)+2.0E0*CR(KR,24)*BN*RI(KR,7)*XX2
      IF(E1R(KR) .EQ. 0.0  ) GO TO 154
      IF(J .LT. I) GO TO 155
C       SUBMATRIX   B
      AK(IN,JN)=AK(IN,JN)+(CR(KR,5)*RI(KR,9)+CR(KR,6)*RI(KR,10)+CR(KR,7)
     1*RI(KR,11)+CR(KR,8)*RI(KR,12))*ABN*XX2
C       SUBMATRIX   Q
      AM(IN,JN)=AM(IN,JN)+(CR(KR,30)*RI(KR,15)+CR(KR,28)*RI(KR,16)+
     1CR(KR,32)*RI(KR,17)+CR(KR,33)*RI(KR,18))*XX2
C       SUBMATRIX   C
      AK(INN,JNN)=AK(INN,JNN)+((CR(KR,5)*RI(KR,9)+CR(KR,6)*RI(KR,10)+
     1CR(KR,8)*RI(KR,12))*ABN2+CR(KR,9)*RI(KR,11)*(AN2+BN2))*XX2+(CR(KR,
     210)*RI(KR,1)*ABN2-CR(KR,4)*RI(KR,2)*(AN2+BN2)+(CR(KR,26)*RI(KR,5)-
     3CR(KR,27)*RI(KR,6))*ABN)*XX1
C       SUBMATRIX   S
      AM(INN,JNN)=AM(INN,JNN)+(CR(KR,30)*RI(KR,4)+CR(KR,34)*ABN*RI(KR,7)
     1)*XX1+(CR(KR,30)*RI(KR,15)+CR(KR,33)*RI(KR,18)+CR(KR,32)*RI(KR,17)
     2)*ABN*XX2
  155 IF(I .LE. 0) GO TO 156
C       SUBMATRIX   E
      AK(I,JNN)=AK(I,JNN)-(CR(KR,4)*ABN2*RI(KR,1)+CR(KR,25)*ABN*RI(KR,5)
     1)*XX1
C       SUBMATRIX   P
      AM(I,JNN)=AM(I,JNN)-(CR(KR,28)*RI(KR,4)+CR(KR,29)*ABN*RI(KR,7))*
     1XX1
C       SUBMATRIX   F
  156 AK(IN,JNN)=AK(IN,JNN)+((CR(KR,8)*RI(KR,12)+CR(KR,5)*RI(KR,9)+CR(KR
     1,6)*RI(KR,10))*ABNB+(ABNB+AN)*CR(KR,9)*RI(KR,11))*XX2
```

```
C        SUBMATRIX  R
         AM(IN,JNN)=AM(IN,JNN)+2.0E0*BN*(CR(KR,31)*RI(KR,16)+CR(KR,30)*RI(K
        1R,15)+CR(KR,33)*RI(KR,18)+CR(KR,32)*RI(KR,17))*XX2
  154 IF(E2R(KR) .EQ. 0.0  ) GO TO 157
        IF(J .LT. I) GO TO 158
C        SUBMATRIX  B
         AK(IN,JN)=AK(IN,JN)+(CR(KR,11)*RI(KR,2)+CR(KR,12)*RI(KR,19)+CR(KR,
        113)*RI(KR,20))*ABN*XX2
C        SUBMATRIX  Q
         AM(IN,JN)=AM(IN,JN)+(CR(KR,35)*R(6)+CR(KR,37)*R(7)+CR(KR,38)*R(2))
        1*XX2
C        SUBMATRIX  C
         AK(INN,JNN)=AK(INN,JNN)+((CR(KR,12)*RI(KR,19)+CR(KR,17)*RI(KR,20))
        1*ABN2+(CR(KR,16)*RI(KR,20)+CR(KR,14)*RI(KR,2))*(AN2+BN2))*XX2
C        SUBMATRIX  S
         AM(INN,JNN)=AM(INN,JNN)+CR(KR,40)*RI(KR,4)*XX1+(CR(KR,40)*R(2)+CR(
        1KR,37)*R(7))*ABN*XX2
  158 IF(I .LE. 0) GO TO 159
C        SUBMATRIX  P
         AM(I,JNN)=AM(I,JNN)-CR(KR,35)*RI(KR,4)*XX1
C        SUBMATRIX  F
  159 AK(IN,JNN)=AK(IN,JNN)+(CR(KR,14)*RI(KR,2)*(ABNB+AN)+(CR(KR,15)*RI(
        1KR,20)+CR(KR,12)*RI(KR,19))*ABNB+CR(KR,16)*RI(KR,20)*AN)*XX2
C        SUBMATRIX  R
         AM(IN,JNN)=AM(IN,JNN)+(CR(KR,35)*R(6)+CR(KR,39)*R(2)+2.0E0*CR(KR,
        137)*R(7))*BN*XX2
  157 IF(IR .EQ. 0) GO TO 144
        IF(J .LT. I) GO TO 160
        IF(J .LE. 0 .OR. I .LE. 0) GO TO 161
C        SUBMATRIX  A
         AK(I,J)=AK(I,J)+CR(KR,1)*(ABN*RI(KR,21)+ABNB*RI(KR,22)+ABNA*RI(KR,
        123))*XX1
C        SUBMATRIX  B
  161 AK(IN,JN)=AK(IN,JN)+CR(KR,3)*(RI(KR,21)+BN*RI(KR,22)+AN*RI(KR,23))
        1*XX2
C        SUBMATRIX  C
         AK(INN,JNN)=AK(INN,JNN)+CR(KR,3)*(ABN*RI(KR,21)+ABNB*RI(KR,22)+ABN
        1A*RI(KR,23))*XX2
  160 IF(I .LE. 0) GO TO 162
C        SUBMATRIX  E
         AK(I,JNN)=AK(I,JNN)+CR(KR,1)*AN*RI(KR,24)*XX1
C        SUBMATRIX  F
  162 AK(IN,JNN)=                AK(IN,JNN)+CR(KR,3)*(BN2*RI(KR,22)+RI(KR,23)
        1*ABN+BN*RI(KR,21))*XX2
        IF(E1R(KR) .EQ. 0.0  ) GO TO 163
        IF(J .LT. I) GO TO 164
C        SUBMATRIX  B
         AK(IN,JN)=AK(IN,JN)+(CR(KR,5)*(RI(KR,43)+BN*RI(KR,35)+AN*RI(KR,36)
        1)+CR(KR,6)*(RI(KR,26)+RI(KR,27)+BN*(RI(KR,37)+RI(KR,39))+AN*(RI(KR
        2,38)+RI(KR,40)))+
        3              CR(KR,8)*(RI(KR,28)+RI(KR,29)+AN*RI(KR,42)+BN*RI(KR,
        441))+CR(KR,9)*(AN*RI(KR,32)+BN*RI(KR,31))+CR(KR,18)*RI(KR,30)+CR(K
        5R,19)*(AN*RI(KR,34)+BN*RI(KR,33)))*XX2
C        SUBMATRIX  C
         AK(INN,JNN)=AK(INN,JNN)+(CR(KR,10)*(RI(KR,21)*ABN+ABNB*RI(KR,22)+
        1ABNA*RI(KR,23))-CR(KR,4)*(BN*RI(KR,25)+AN*RI(KR,24)))*XX1+((CR(KR,
        25)*RI(KR,43)+CR(KR,6)*(RI(KR,26)+RI(KR,27))+CR(KR,18)*RI(KR,30)+CR
        3(KR,8)*(RI(KR,29)+RI(KR,28)))*ABN+ CR(KR,5)*(ABNB*RI(KR,35)+ABNA*
        4RI(KR,36))+CR(KR,6)*(ABNB*(RI(KR,37)+RI(KR,39))+ABNA*(RI(KR,38)+
        5RI(KR,40)))+CR(KR,8)*(ABNB*RI(KR,41)+ABNA*RI(KR,42))+CR(KR,19)*(AB
        6NB*RI(KR,33)+ABNA*RI(KR,34))+CR(KR,9)*(BN*RI(KR,32)+AN*RI(KR,31)))
        7*XX2
```

```
  164 IF( I .LE. 0) GO TO 165
C       SUBMATRIX  E
      AK(I,JNN)=AK(I,JNN)-CR(KR,4)*(ABN*RI(KR,21)+ABNA*RI(KR,23)+ABNB*
     1RI(KR,22))*XX1
C       SUBMATRIX  F
  165 AK(IN,JNN)=AK(IN,JNN)+((CR(KR,5)*RI(KR,35)+
     *                                                   CR(KR,6)*(RI(KR,37)+RI(K
     1R,39))+CR(KR,8)*RI(KR,41)+CR(KR,19)*RI(KR,33))*BN2+(CR(KR,9)*RI(KR
     2,32)+CR(KR,5)*RI(KR,36)+CR(KR,19)*RI(KR,34)+CR(KR,6)*(RI(KR,38)+RI
     3(KR,40))+CR(KR,8)*RI(KR,42))*ABN+(CR(KR,5)*RI(KR,43)+CR(KR,6)*(RI(
     4KR,26)+RI(KR,27))+CR(KR,8)*(RI(KR,28)+RI(KR,29))+CR(KR,18)*RI(KR,3
     50))*BN+CR(KR,9)*RI(KR,31))*XX2
  163 IF(E2R(KR) .EQ. 0.0  ) GO TO 144
      IF(J .LT. I) GO TO 166
C       SUBMATRIX  B
      AK(IN,JN)=AK(IN,JN)+(CR(KR,12)*(RI(KR,44)+RI(KR,46)+BN*RI(KR,49)
     1+AN*RI(KR,50))+CR(KR,17)*(RI(KR,45)+BN*(RI(KR,47)+RI(KR,53))+AN*(
     3RI(KR,48)+RI(KR,54)))+CR(KR,14)*(BN*RI(KR,24)+AN*RI(KR,25))+CR(KR,
     420)*(BN*RI(KR,51)+AN*RI(KR,52)))*XX2
C       SUBMATRIX  C
      AK(INN,JNN)=AK(INN,JNN)+(CR(KR,12)*(ABN*(RI(KR,44)+RI(KR,46))+ABNB
     1*RI(KR,49)+ABNA*RI(KR,50))+CR(KR,17)*RI(KR,45)*ABN+CR(KR,20)*(ABNB
     2*RI(KR,51)+ABNA*RI(KR,52))+CR(KR,16)*(ABNB*(RI(KR,53)+RI(KR,47))+
     4ABNA*(RI(KR,54)+RI(KR,48))+BN*(RI(KR,48)+RI(KR,54))+AN*(RI(KR,47)+
     5RI(KR,53)))+CR(KR,14)*(AN*RI(KR,24)+BN*RI(KR,25)))*XX2
C       SUBMATRIX  F
  166 AK(IN,JNN)=AK(IN,JNN)+((CR(KR,12)*RI(KR,49)+CR(KR,20)*RI(KR,51)+
     1CR(KR,16)*(RI(KR,53)+RI(KR,47)))*BN2+(CR(KR,17)*(RI(KR,48)+RI(KR,
     254))+CR(KR,12)*RI(KR,50)+CR(KR,14)*RI(KR,25)+CR(KR,20)*RI(KR,52))
     3*ABN+(CR(KR,12)*(RI(KR,44)+RI(KR,46))+CR(KR,17)*RI(KR,45))*BN+CR(
     4KR,16)*(RI(KR,47)+RI(KR,53))+CR(KR,14)*RI(KR,24))*XX2
  144 CONTINUE
   93 CONTINUE
   92 CONTINUE
   91 CONTINUE
   90 CONTINUE
C
C     COMPUTE THE LOWER DIAGONAL ELEMENTS OF THE SYMMETRIC MASS AND
C     STIFFNESS MATRICES
C
      DO 2002 I=2,MN3
      I1=I-1
      DO 2002 J=1,I1
      AK(I,J)=AK(J,I)
 2002 AM(I,J)=AM(J,I)
      IF(NWK-1)2010,2011,2012
 2012 WRITE(7,65)((AK(I,J),J=1,MN3),I=1,MN3)
 2011 WRITE(6,2015)
 2015 FORMAT(1H1,30X,17HSTIFFNESS  MATRIX,/,31X,17(1H=),//)
      DO 2013 I=1,MN3
      WRITE(6,2014)(AK(I,J),J=1,MN3)
 2014 FORMAT(5X,E15.8,1X,E15.8,1X,E15.8,1X,E15.8)
 2013 CONTINUE
 2010 IF(NWM-1)2016,2017,2018
 2018 WRITE(7,65)((AM(I,J),J=1,MN3),I=1,MN3)
 2017 WRITE(6,2019)
 2019 FORMAT(1H1,32X,12HMASS  MATRIX,/,33X,12(1H=),//)
      DO 2020 I=1,MN3
      WRITE(6,2014)(AM(I,J),J=1,MN3)
 2020 CONTINUE
C
C     EVALUATE THE EIGENVALUES AND EIGENVECTORS
```

```
C
 2016 CALL EIGEND (AM,AK,MN3,KRRR,VECR,LC,XXXX,Y,MC,Z,EVR,EVI,INDIC)
C
C      CONVERT THE EIGENVALUES INTO FREQUENCIES IN HERTZ
C
      DO 2006 I=1,MN3
      IF(EVR(I) .LE. 0.0  ) GO TO 2006
      EVR(I)= SQRT(EVR(I))/PI2
 2006 CONTINUE
      IF(NWEV-1)2021,2022,2023
 2023 DO 2024 J=1,MN3
      WRITE(7,65)(VECR(I,J), I=1,MN3)
 2024 CONTINUE
 2022 WRITE(6,2025)
 2025 FORMAT(1H1,32X,12HEIGENVECTORS,/,33X,12(1H=),///)
      DO 2026 J=1,MN3
      WRITE(6,2014)(VECR(I,J),I=1,MN3)
 2026 CONTINUE
 2021 WRITE(6,2027)
 2027 FORMAT(1H1,27X,22HEIGENVALUES  IN  HERTZ,/,28X,22(1H=),///)
      WRITE(6,2014)(EVR(I),I=1,MN3)
      GO TO 10000
   99 WRITE(6,100)
  100 FORMAT(///,1X,62H***** ERROR *****      MMIN = 0 IS POSSIBLE ONLY WI
     1TH FREE-FREE,/,32X,18HBOUNDARY CONDITION)
      GO TO 10000
10000 CONTINUE
      CALL EXIT
      END
```

```
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                    +
C     + SUBROUTINE INTGRL                                                  +
C     +                                                                    +
C     + PURPOSE                                                            +
C     +    TO EVALUATE THE LONGITUDINAL INTEGRALS XI(1) TO XI(5) WHEN      +
C     +    M, MB, AA, AND NBC ARE FURNISHED THROUGH THE COMMON STATEMENT   +
C     +                                                                    +
C     + USAGE                                                              +
C     +    CALL INTGRL                                                     +
C     +                                                                    +
C     + DESCRIPTION OF THE PARAMETERS                                      +
C     +    M           M                                                   +
C     +    MB          MB                                                  +
C     +    AA          LENGTH OF THE SHELL                                 +
C     +    NBC         CODE NUMBER OF THE BOUNDARY CONDITION WHICH IS      +
C     +                UNDER CONSIDERATION                                 +
C     +    PI          3.1415926535                                        +
C     +    XI(I)       LONGITUDINAL INTEGRAL                               +
C     +                                                                    +
C     + SUBROUTINES REQUIRED                                               +
C     +    NONE                                                            +
C     +                                                                    +
C     + METHOD                                                             +
C     +    THE CLOSE FORM EXPRESSIONS FOR THE INTEGRALS WERE OBTAINED      +
C     +    FROM  ROBERT P. FELGAR, JR., FORMULAS FOR INTEGRALS             +
C     +    CONTAINING CHARACTERISTICFUNCTIONS OF A VIBRATING BEAM,         +
C     +    CIRCULAR NUMBER 14, THE UNIVERSITY OF TEXAS, 1950               +
C     +                                                                    +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE INTGRL
      DIMENSION A(5),B(5)
      COMMON DR(9),R(9),DRV(5),RV(5),R1(8),RR1(8),R2(10),RR2(10),R3(2),
     1RR3(2),R4(5),RR4(5),R5(18),RR5(18),R6(11),RR6(11),PI,XK,AA,XI(5),
     2XR(2),E1,E2,N,NB,NBC,M,MB,NSA
      DO 8 I=1,5
      XI(I)=0.0
    8 CONTINUE
      GO TO (100,200,300,400),NBC
C     CLAMPED-FREE
C     ALPHAS AND BETAS OBTAINED FROM FELGAR AND YOUNG
  100 A(1)=0.7340955
      A(2)=1.01846644
      A(3)=0.99922450
      A(4)=1.000033553
      A(5)=0.9999985501
      B(1)=1.8751041E0/AA
      B(2)=4.69409113E0/AA
      B(3)=7.85475743E0/AA
      B(4)=10.99554074E0/AA
      B(5)=14.13716839E0/AA
      IF(M - 5) 15,15,25
   15 AM=A(M)
      BM=B(M)
      GO TO 30
   25 BM=(2.0E0* FLOAT(M)-1.0E0)*PI/(2.0E0*AA)
      AM=1.0
   30 AB=AM*BM
      B4=BM*BM*BM*BM
      IF(MB - 5)40,40,50
   40 AMB=A(MB)
      BMB=B(MB)
```

```
      GO TO 20
   50 AMB=1.0
      BMB=(2.0E0* FLOAT(MB)-1.0E0)*PI/(2.0E0*AA)
   20 ABB=AMB*BMB
      B4B=BMB*BMB*BMB*BMB
      IF(M-MB) 60,65,60
C       M  =  MB
   65 XI(1)=B4*AA
      XI(2)=AB*(2.0   +AB*AA)
      XI(3)=AB*(2.0   -AB*AA)
      XI(4)=XI(3)
      XI(5)=AA
      RETURN
C       M  NOT  =  MB
   60 BBM=BM*BMB
      MPMB=M+MB
      A3B=AMB*BM*BM*BM
      AB3=AM*BMB*BMB*BMB
      XI(2)=4.0E0*BBM*((-1.0E0)**MPMB*(A3B-AB3)-BBM*(AB-ABB))/(B4-B4B)
      XI(3)=4.0E0*BM*BM*(ABB-AB)*(((-1.0E0)**MPMB)*BM*BM+BMB*BMB)/
     1(B4B-B4)
      XI(4)=4.0E0*BMB*BMB*(AB-ABB)*(((-1.0E0)**MPMB)*BMB*BMB+BM*BM)/
     1(B4-B4B)
      RETURN
C     FREELY SUPPORTED
  200 IF(M-MB)70,75,70
C       M  =  MB
   75 BM=M
      XI(2)=BM*BM*PI*PI/AA
      XI(3)=-XI(2)
      XI(4)=XI(3)
      XI(1)=XI(2)*XI(2)/AA
      XI(5)=AA
   70 RETURN
C     CLAMPED CLAMPED
  300 A(1)=0.9825022158
      A(2)=1.000777311
      A(3)=0.9999664501
      A(4)=1.000001450
      A(5)=0.9999999373
      B(1)=4.7300408E0/AA
      B(2)=7.8532046E0/AA
      B(3)=10.9956078E0/AA
      B(4)=14.1371655E0/AA
      B(5)=17.2787596E0/AA
      IF(M - 5)1,1,2
    1 AM=A(M)
      BM=B(M)
      GO TO 3
    2 BM=(2.0E0* FLOAT(M)+1.0E0)*PI/(2.0E0*AA)
      AM=1.0
    3 IF(MB - 5)4,4,5
    4 AMB=A(MB)
      BMB=B(MB)
      GO TO 6
    5 AMB=1.0
      BMB=(2.0E0* FLOAT(MB) +1.0E0)*PI/(2.0E0*AA)
    6 AB=AM*BM
      ABB=AMB*BMB
      B4=BM*BM*BM*BM
      B4B=BMB*BMB*BMB*BMB
      BB2=BM*BMB*BM*BMB
```

```
      IF(M-MB)80,85,80
C        M  =  MB
   85 XI(1)=B4*AA
      XI(2)=AB*(AB*AA-2.0   )
      XI(5)=AA
      GO TO 90
C        M  NOT  =  MB
   80 XI(2)=4.0E0*BB2*(ABB-AB)*((-1.0E0)**(M+MB)+1.0E0)/(B4-B4B)
   90 XI(3)=-XI(2)
      XI(4)=XI(3)
      RETURN
C     FREE-FREE
C     ALPHAS AND BETAS OBTAINED FROM     FELGAR AND YOUNG
  400 A(1)=0.9825022158
      A(2)=1.0007773111
      A(3)=0.9999664501
      A(4)=1.000001450
      A(5)=0.9999999373
      B(1)=4.7300408E0/AA
      B(2)=7.8532046E0/AA
      B(3)=10.9956078E0/AA
      B(4)=14.1371655E0/AA
      B(5)=17.2787596E0/AA
      MM1=M-1
      MBM1=MB-1
      M2=M+MB-2
      IF(M .LT. 2) GO TO 1730
      IF(M - 6) 715,715,725
  715 AM=A(MM1)
      BM=B(MM1)
      GO TO 730
  725 BM=(2.0E0* FLOAT(MM1)+1.0E0)*PI/(2.0E0*AA)
      AM=1.0
  730 AB=AM*BM
      B4=BM*BM*BM*BM
 1730 IF(MB .LT. 2) GO TO 1720
      IF(MB - 6)740,740,750
  740 AMB=A(MBM1)
      BMB=B(MBM1)
      GO TO 720
  750 AMB=1.0
      BMB=(2.0E0* FLOAT(MBM1)+1.0E0)*PI/(2.0E0*AA)
  720 ABB=AMB*BMB
      B4B=BMB*BMB*BMB*BMB
 1720 CONTINUE
      S1=(1.0  -(-1.0   )**MBM1)
      S2=(1.0  +(-1.0   )**MBM1)
      S3=(1.0  -(-1.0   )**MM1)
      S4=(1.0  +(-1.0   )**MM1)
      S5=(1.0  +(-1.0   )**M2)
      IF(MM1)150,250,350
  150 IF(MB .NE. M) GO TO 151
C     M=0  MB=0
      XI(5)=AA
      RETURN
  151 IF(MB .LT. 2) RETURN
C        M=0   MB GREATER THAN OR =2
      XI(4)=2.0E0*ABB*S1
      RETURN
  250 IF(MB .NE. M) GO TO 251
C     M=1  MB=1
      XI(2)=1.0E0/AA
```

```
      XI(5)=AA/12.0
      RETURN
  251 IF(MB .LT. 2) RETURN
C      M=1   MB GREATER THAN OR =2
      XI(2)=-2.0E0*S2/AA
      XI(4)=(2.0F0/AA-ABB)*S2
      RETURN
  350 IF(MBM1)450,550,650
C      M  GREATER THAN OR =2  MB=0
  450 XI(3)=2.0E0*AB*S3
      RETURN
C      M  GREATER THAN OR =2  MB=1
  550 XI(2)=-2.0E0*S4/AA
      XI(3)=S4*(2.0E0/AA-AB)
      RETURN
C      M  AND  MB  GREATER THAN OR =2
  650 IF(M .NE. MB) GO TO 651
C     M = MB
      XI(1)=AA*B4
      XI(2)=AB*(AB*AA+6.0E0)
      XI(3)=AB*(2.0E0-AB*AA)
      XI(4)=XI(3)
      XI(5)=AA
      RETURN
C     M  NOT = MB
  651 AB3=AM*BMB*BMB*BMB
      A3B=AMB*BM*BM*BM
      XI(2)=4.0E0*BM*BMB*(AB3-A3B)*S5/(B4B-B4)
      XI(3)=4.0E0*B4*(ABB-AB)*S5/(B4B-B4)
      XI(4)=4.0E0*B4B*(AB-ABB)*S5/(B4-B4B)
      RETURN
      END
```

```
C    *****************************************************************
C    *                                                               *
C    * SUBROUTINE XX                                                 *
C    *                                                               *
C    * PURPOSE                                                       *
C    *    CALCULATE THE VALUES OF XR(1) AND, XR(2) AT A GIVEN VALUE OF *
C    *    X = XK                                                     *
C    *                                                               *
C    * USAGE                                                         *
C    *    CALL XX                                                    *
C    *                                                               *
C    * DESCRIPTION OF THE PARAMETERS                                 *
C    *    PI          3.1415926535                                   *
C    *    XK          THE VALUE OF X AT WHICH THE FUNCTIONS XR(1) AND *
C    *                XR(2) ARE TO BE EVALUATED                      *
C    *    AA          LENGTH OF THE SHELL                            *
C    *    NBC         THE CODE NUMBER OF THE BOUNDARY CONDITION UNDER *
C    *                CONSIDERATION                                  *
C    *    M           M                                              *
C    *    MB          MB                                             *
C    *                                                               *
C    * REMARKS                                                       *
C    *    THE INPUT AND OUTPUT PARAMETERS ARE COMMUNICATED THROUGH THE *
C    *    COMMON STATEMENT                                           *
C    *                                                               *
C    * SUBROUTINES REQUIRED                                          *
C    *    NONE                                                       *
C    *                                                               *
C    * METHOD                                                        *
C    *    THE ASSUMED AXIAL MODE FUNCTIONS ARE GENERATED IN THIS     *
C    *    SUBROUTINE USING FELGAR AND YOUNG'S BEAM FUNCTIONS         *
C    *                                                               *
C    *****************************************************************
      SUBROUTINE XX
C     THIS SUBROUTINE EVALUATES THE VALUES OF  XR(1), XR(2)  AT A GIVEN
C     VALUE OF  X
      DIMENSION A(5),B(5)
      COMMON DR(9),R(9),DRV(5),RV(5),R1(8),RR1(8),R2(10),RR2(10),R3(2),
     1RR3(2),R4(5),RR4(5),R5(18),RR5(18),R6(11),RR6(11),PI,XK,AA,XI(5),
     2XR(2),E1,E2,     N,NB,NBC,M,MB,NSA
      XR(1)=0.0
      XR(2)=0.0
      GO TO (100,200,300,400),NBC
C      CLAMPED-FREE
C     ALPHAS AND BETAS OBTAINED FROM FELGAR AND YOUNG
  100 A(1)=0.7340955
      A(2)=1.01846644
      A(3)=0.99922450
      A(4)=1.00033553
      A(5)=0.9999985501
      B(1)=1.8751041E0/AA
      B(2)=4.69409113E0/AA
      B(3)=7.85475743E0/AA
      B(4)=10.99554074E0/AA
      B(5)=14.13716839E0/AA
      IF(M - 5) 15,15,25
   15 AM=A(M)
      BM=B(M)
      GO TO 30
   25 BM=(2.0E0* FLOAT(M)-1.0E0)*PI/(2.0E0*AA)
      AM=1.0
   30 IF(MB - 5)40,40,50
```

```
   40 AMB=A(MB)
      BMB=B(MB)
      GO TO 20
   50 AMB=1.0
      BMB=(2.0E0* FLOAT(MB)-1.0E0)*PI/(2.0E0*AA)
   20 BMX=BM*XK
      BMBX=BMB*XK
      EB= EXP(BMX)
      EBB= EXP(BMBX)
      S= SIN(BMX)
      SB= SIN(BMBX)
      C= COS(BMX)
      CB= COS(BMBX)
      SH=(EB-1.0E0/EB)/2.0E0
      SHB=(EBB-1.0E0/EBB)/2.0E0
      CH=(EB+1.0E0/EB)/2.0E0
      CHB=(EBB+1.0E0/EBB)/2.0E0
      XR(1)=BM*BMB*(SH+S-AM*(CH-C))*(SHB+SB-AMB*(CHB-CB))
      XR(2)=(CH-C-AM*(SH-S))*(CHB-CB-AMB*(SHB-SB))
      RETURN
C     FREELY SUPPORTED
  200 BM=M
      BMB=MB
      PIOA=PI/AA
      XM=PIOA*BM*XK
      XMB=PIOA*BMB*XK
      XR(1)=2.0E0*BM*BMB*PIOA*PIOA* COS(XM)* COS(XMB)
      XR(2)=2.0E0* SIN(XM)* SIN(XMB)
      RETURN
C     CLAMPED CLAMPED
  300 A(1)=0.9825022158
      A(2)=1.000777311
      A(3)=0.9999664501
      A(4)=1.000001450
      A(5)=0.9999999373
      B(1)=4.7300408E0/AA
      B(2)=7.8532046E0/AA
      B(3)=10.9956078E0/AA
      B(4)=14.1371655E0/AA
      B(5)=17.2787596E0/AA
      IF(M - 5)1,1,2
    1 AM=A(M)
      BM=B(M)
      GO TO 3
    2 BM=(2.0E0* FLOAT(M)+1.0E0)*PI/(2.0E0*AA)
      AM=1.0
    3 IF(MB - 5)4,4,5
    4 AMB=A(MB)
      BMB=B(MB)
      GO TO 20
    5 AMB=1.0
      BMB=(2.0E0* FLOAT(MB) +1.0E0)*PI/(2.0E0*AA)
      GO TO 20
C     FREE-FREE
  400 A(1)=0.9825022158
      A(2)=1.000777311
      A(3)=0.9999664501
      A(4)=1.000001450
      A(5)=0.9999999373
      B(1)=4.7300408E0/AA
      B(2)=7.8532046E0/AA
      B(3)=10.9956078E0/AA
```

```
      B(4)=14.1371655E0/AA
      B(5)=17.2787596E0/AA
      MM1=M-1
      MBM1=MB-1
      IF(M .LT. 2) GO TO 1730
      IF(M - 6) 715,715,725
  715 AM=A(MM1)
      BM=B(MM1)
      GO TO 1740
  725 BM=(2.0E0* FLOAT(MM1)+1.0E0)*PI/(2.0E0*AA)
      AM=1.0
 1740 BMX=BM*XK
      S= SIN(BMX)
      C= COS(BMX)
      EB= EXP(BMX)
      SH=(EB-1.0E0/EB)/2.0E0
      CH=(EB+1.0E0/EB)/2.0E0
 1730 IF(MB .LT. 2) GO TO 1720
      IF(MB - 6)740,740,750
  740 AMB=A(MBM1)
      BMB=B(MBM1)
      GO TO 1760
  750 AMB=1.0
      BMB=(2.0E0* FLOAT(MBM1)+1.0E0)*PI/(2.0E0*AA)
 1760 BMBX=BMB*XK
      SB= SIN(BMBX)
      CB= COS(BMBX)
      EBB= EXP(BMBX)
      SHB=(EBB-1.0E0/EBB)/2.0E0
      CHB=(EBB+1.0E0/EBB)/2.0E0
 1720 IF(MM1) 150,250,350
  150 IF(MB .NE. M) GO TO 151
C       M=0  MB=0
      XR(2)=1.0
      RETURN
  151 IF(MBM1) 152,152,153
C       M=0  MB=1
  152 XR(2)=XK/AA-0.5
      RETURN
C       M=0    MB GREATER THAN OR =2
  153 XR(2)=CHB+CB-AMB*(SHB+SB)
      RETURN
  250 IF(MBM1) 251,252,253
C       M=1  MB=0
  251 XR(2)=XK/AA-0.5
      RETURN
C       M=1  MB=1
  252 XR(1)=1.0E0/(AA*AA)
      XR(2)=XK*XK/(AA*AA)+0.25   -XK/AA
      RETURN
C       M=1    MB GREATER THAN OR =2
  253 XR(1)=BMB*(SHB-SB-AMB*(CHB+CB))/AA
      XR(2)=(XK/AA-0.5   )*(CHB+CB-AMB*(SHB+SB))
      RETURN
  350 IF(MBM1) 351,352,353
C       M  GREATER THAN OR EQUAL TO 2  MB=0
  351 XR(2)=CH+C-AM*(SH +S)
      RETURN
C       M  GREATER THAN OR EQUAL TO 2  MB=1
  352 XR(1)=BM*(SH-S-AM*(CH+C))/AA
      XR(2)=(XK/AA-0.5   )*(CH+C-AM*(SH+S))
      RETURN
```

```
C        M  AND  MB  GREATER THAN OR =2
  353 XR(1)=BM*BMB*(SH-S-AM*(CH+C))*(SHB-SB-AMB*(CHB+CB))
      XR(2)=(CH+C-AM*(SH+S))*(CHB+CB-AMB*(SHB+SB))
      RETURN
      END
```

```
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                        +
C     + SUBROUTINE SHELL1(X)                                                   +
C     +                                                                        +
C     + PURPOSE                                                                +
C     +    CALCULATE AT A GIVEN VALUE OF X THE VALUES OF THE INTEGRANDS       +
C     +    OF THE FIRST SET OF CIRCUMFERENTIAL INTEGRALS OF THE SHELL         +
C     +                                                                        +
C     + USAGE                                                                  +
C     +    IT IS USED AS AN ARGUMENT OF THE NUMERICAL INTEGRATION             +
C     +    SUBROUTINE  'GAUSS'                                                 +
C     +                                                                        +
C     + DESCRIPTION OF THE PARAMETERS                                          +
C     +    X           THE INPUT VALUE OF X AT WHICH THE INTEGRANDS HAVE      +
C     +                TO BE EVALUATED                                         +
C     +    DR(I)       INTEGRANDS                                              +
C     +    N           N                                                       +
C     +    NB          NB                                                      +
C     +    NSA         0 WHEN COMPUTING THE SYMMETRIC  MODE SHAPES            +
C     +                1 WHEN COMPUTING THE ANTISYMMETRIC MODE SHAPES         +
C     +                                                                        +
C     + REMARKS                                                                +
C     +    EXCEPT FOR X ALL THE INPUT AND OUTPUT PARAMETERS ARE               +
C     +    COMMUNICATED THROUGH THE COMMON STATEMENT                          +
C     +                                                                        +
C     + FUNCTION SUBROUTINES REQUIRED                                          +
C     +    RSHL                                                                +
C     +                                                                        +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE SHELL1(X)
      COMMON DR(9),R(9),DRV(5),RV(5),R1(8),RR1(8),R2(10),RR2(10),R3(2),
     1RR3(2),R4(5),RR4(5),R5(18),RR5(18),R6(11),RR6(11),PI,XK,AA,XI(5),
     2XR(2),E1,E2,N,NB,NBC,M,MB,NSA
      XN=X*N
      XNB=X*NB
      SR=RSHL(X)
      IF(NSA)1,1,2
    1 CN= COS(XN)
      SN= SIN(XN)
      CNB= COS(XNB)
      SNB= SIN(XNB)
      GO TO 3
    2 CN= SIN(XN)
      SN= COS(XN)
      CNB= SIN(XNB)
      SNB= COS(XNB)
    3 DR(5)=CN*CNB
      DR(6)=SN*SNB
      DR(1)=SR*DR(5)
      DR(9)=SR*DR(6)
      DR(2)=DR(6)/SR
      DR(8)=DR(5)/SR
      DR(7)=DR(2)/SR
      DR(3)=DR(7)/SR
      DR(4)=DR(8)/(SR*SR)
      RETURN
      END
```

```
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                    +
C     + SUBROUTINE SHELL2(X)                                               +
C     +                                                                    +
C     + PURPOSE                                                            +
C     +    CALCULATE AT A GIVEN VALUE OF X THE VALUES OF THE INTEGRANDS    +
C     +    OF THE SECOND SET OF CIRCUMFERENTIAL INTEGRALS OF THE SHELL     +
C     +                                                                    +
C     + USAGE                                                              +
C     +    IT IS USED AS AN ARGUMENT OF THE NUMERICAL INTEGRATION          +
C     +    SUBROUTINE  *GAUSS*                                             +
C     +                                                                    +
C     + DESCRIPTION OF THE PARAMETERS                                      +
C     +    X          THE INPUT VALUE OF X AT WHICH THE INTEGRANDS HAVE    +
C     +               TO BE EVALUATED                                      +
C     +    DRV(I)     INTEGRANDS                                           +
C     +    N          N                                                    +
C     +    NB         NB                                                   +
C     +    NSA        0 WHEN COMPUTING THE SYMMETRIC MODE SHAPES           +
C     +               1 WHEN COMPUTING THE ANTISYMMETRIC MODE SHAPES       +
C     +                                                                    +
C     + REMARKS                                                            +
C     +    EXCEPT FOR X ALL THE INPUT AND OUTPUT PARAMETERS ARE            +
C     +    COMMUNICATED THROUGH THE COMMON STATEMENT                       +
C     +                                                                    +
C     + FUNCTION SUBROUTINES REQUIRED                                      +
C     +    RSHL                                                            +
C     +    RRRT                                                            +
C     +                                                                    +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE SHELL2(X)
      COMMON DR(9),R(9),DRV(5),RV(5),R1(8),RR1(8),R2(10),RR2(10),R3(2),
     1RR3(2),R4(5),RR4(5),R5(18),RR5(18),R6(11),RR6(11),PI,XK,AA,XI(5),
     2XR(2),E1,E2,N,NB,NBC,M,MB,NSA
      XN=X*N
      XNB=X*NB
      SR=RSHL(X)
      RT=RRRT(X)
      RT2=RT*RT
      SR2=1.0E0/(SR*SR)
      IF(NSA)1,1,2
    1 SN= SIN(XN)
      SNB= SIN(XNB)
      CNB= COS(XNB)
      CN= COS(XN)
      GO TO 3
    2 SN= COS(XN)
      SNB= COS(XNB)
      CNB= SIN(XNB)
      CN= SIN(XN)
      RT=-RT
    3 DRV(1)=RT2*SN*SNB/SR
      DRV(5)=RT*CN*SNB
      DRV(4)=DRV(5)*SR2
      RETURN
      END
```

```
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                        +
C     + SUBROUTINE RING1(X)                                                    +
C     +                                                                        +
C     + PURPOSE                                                                +
C     +    CALCULATE AT A GIVEN VALUE OF X THE VALUES OF THE INTEGRANDS        +
C     +    OF THE FIRST SET OF CIRCUMFERENTIAL INTEGRALS OF THE RING           +
C     +                                                                        +
C     + USAGE                                                                  +
C     +    IT IS USED AS AN ARGUMENT OF THE NUMERICAL INTEGRATION              +
C     +    SUBROUTINE  'GAUSS'                                                 +
C     +                                                                        +
C     + DESCRIPTION OF THE PARAMETERS                                          +
C     +    E1         Z-DISTANCE OF THE SHEAR CENTER OF THE RING FROM THE      +
C     +               MIDDLE SURFACE OF THE SHELL                              +
C     +    E2         Z-DISTANCE OF THE CENTROID OF THE RING FROM ITS          +
C     +               SHEAR CENTER                                             +
C     +    X          THE INPUT VALUE OF X AT WHICH THE INTEGRANDS HAVE        +
C     +               TO BE EVALUATED                                          +
C     +    R1(I)      INTEGRANDS                                               +
C     +    N          N                                                        +
C     +    NB         NB                                                       +
C     +    NSA        0 WHEN COMPUTING THE SYMMETRIC  MODE SHAPES              +
C     +               1 WHEN COMPUTING THE ANTISYMMETRIC MODE SHAPES           +
C     +                                                                        +
C     + REMARKS                                                                +
C     +    EXCEPT FOR X ALL THE INPUT AND OUTPUT PARAMETERS ARE                +
C     +    COMMUNICATED THROUGH THE COMMON STATEMENT                           +
C     +                                                                        +
C     + FUNCTION SUBROUTINES REQUIRED                                          +
C     +    RSHL                                                                +
C     +                                                                        +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE RING1(X)
      COMMON DR(9),R(9),DRV(5),RV(5),R1(8),RR1(8),R2(10),RR2(10),R3(2),
     1RR3(2),R4(5),RR4(5),R5(18),RR5(18),R6(11),RR6(11),PI,XK,AA,XI(5),
     2XR(2),E1,E2,N,NB,NBC,M,MB,NSA
      XN=X*N
      XNB=X*NB
      SR=RSHL(X)
      RC=SR+E1+E2
      IF(NSA)1,1,2
    1 CN= COS(XN)
      SN= SIN(XN)
      CNB= COS(XNB)
      SNB= SIN(XNB)
      GO TO 3
    2 CN= SIN(XN)
      SN= COS(XN)
      CNB= SIN(XNB)
      SNB= COS(XNB)
    3 CC=CN*CNB
      SS=SN*SNB
      R1(4)=RC*CC
      R1(8)=RC*SS
      R1(3)=CC/RC
      R1(7)=SS/RC
      R1(2)=R1(3)/RC
      R1(6)=R1(7)/RC
      R1(1)=R1(2)/RC
      R1(5)=R1(6)/RC
      RETURN
      END
```

```
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                      +
C     + SUBROUTINE RING2(X)                                                  +
C     +                                                                      +
C     + PURPOSE                                                              +
C     +    CALCULATE AT A GIVEN VALUE OF X THE VALUES OF THE INTEGRANDS      +
C     +    OF THE SECOND SET OF CIRCUMFERENTIAL INTEGRALS OF THE RING        +
C     +                                                                      +
C     + USAGE                                                                +
C     +    IT IS USED AS AN ARGUMENT OF THE NUMERICAL INTEGRATION            +
C     +    SUBROUTINE  'GAUSS'                                               +
C     +                                                                      +
C     + DESCRIPTION OF THE PARAMETERS                                        +
C     +    E1         Z-DISTANCE OF THE SHEAR CENTER OF THE RING FROM THE   +
C     +               MIDDLE SURFACE OF THE SHELL                            +
C     +    E2         Z-DISTANCE OF THE CENTROID OF THE RING FROM ITS        +
C     +               SHEAR CENTER                                           +
C     +    X          THE INPUT VALUE OF X AT WHICH THE INTEGRANDS HAVE      +
C     +               TO BE EVALUATED                                        +
C     +    R2(I)      INTEGRANDS                                             +
C     +    N          N                                                      +
C     +    NB         NB                                                     +
C     +    NSA        0 WHEN COMPUTING THE SYMMETRIC  MODE SHAPES            +
C     +               1 WHEN COMPUTING THE ANTISYMMETRIC MODE SHAPES         +
C     +                                                                      +
C     + REMARKS                                                              +
C     +    EXCEPT FOR X ALL THE INPUT AND OUTPUT PARAMETERS ARE              +
C     +    COMMUNICATED THROUGH THE COMMON STATEMENT                         +
C     +                                                                      +
C     + FUNCTION SUBROUTINES REQUIRED                                        +
C     +    RSHL                                                              +
C     +                                                                      +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE RING2(X)
     COMMON DR(9),R(9),DRV(5),RV(5),R1(8),RR1(8),R2(10),RR2(10),R3(2),
    1RR3(2),R4(5),RR4(5),R5(18),RR5(18),R6(11),RR6(11),PI,XK,AA,XI(5),
    2XR(2),E1,E2,N,NB,NBC,M,MB,NSA
      XN=X*N
      XNB=X*NB
      SR=RSHL(X)
      RC=SR+E1+E2
      IF(NSA)1,1,2
    1 CN= COS(XN)
      SN= SIN(XN)
      CNB= COS(XNB)
      SNB= SIN(XNB)
      GO TO 3
    2 CN= SIN(XN)
      SN= COS(XN)
      CNB= SIN(XNB)
      SNB= COS(XNB)
    3 CC=CN*CNB
      SS=SN*SNB
      RC2=RC*RC
      RCSR=RC*SR
      R2(10)=SS/RCSR
      R2(9)=R2(10)/SR
      R2(8)=RC*SS/SR
      R2(7)=R2(8)/SR
      R2(6)=SS/RC2
      R2(5)=R2(6)/RC
      R2(3)=CC/RCSR
```

```
R2(1)=R2(3)/SR
R2(4)=R2(3)/RC2
R2(2)=R2(4)/SR
RETURN
END
```

```
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                  +
C     + SUBROUTINE RING3(X)                                              +
C     +                                                                  +
C     + PURPOSE                                                          +
C     +    CALCULATE AT A GIVEN VALUE OF X THE VALUES OF THE INTEGRANDS  +
C     +    OF THE THIRD  SET OF CIRCUMFERENTIAL INTEGRALS OF THE RING    +
C     +                                                                  +
C     + USAGE                                                            +
C     +    IT IS USED AS AN ARGUMENT OF THE NUMERICAL INTEGRATION        +
C     +    SUBROUTINE  'GAUSS'                                           +
C     + DESCRIPTION OF THE PARAMETERS                                    +
C     +    E1          Z-DISTANCE OF THE SHEAR CENTER OF THE RING FROM THE +
C     +                MIDDLE SURFACE OF THE SHELL                       +
C     +    E2          Z-DISTANCE OF THE CENTROID OF THE RING FROM ITS   +
C     +                SHEAR CENTER                                      +
C     +    X           THE INPUT VALUE OF X AT WHICH THE INTEGRANDS HAVE +
C     +                TO BE EVALUATED                                   +
C     +    R3(I)       INTEGRANDS                                        +
C     +    N           N                                                 +
C     +    NB          NB                                                +
C     +    NSA         0 WHEN COMPUTING THE SYMMETRIC  MODE SHAPES       +
C     +                1 WHEN COMPUTING THE ANTISYMMETRIC MODE SHAPES    +
C     +                                                                  +
C     + REMARKS                                                          +
C     +    EXCEPT FOR X ALL THE INPUT AND OUTPUT PARAMETERS ARE          +
C     +    COMMUNICATED THROUGH THE COMMON STATEMENT                     +
C     +                                                                  +
C     + FUNCTION SUBROUTINES REQUIRED                                    +
C     +    RSHL                                                          +
C     +                                                                  +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE RING3(X)
      COMMON DR(9),R(9),DRV(5),RV(5),R1(8),RR1(8),R2(10),RR2(10),R3(2),
     1RR3(2),R4(5),RR4(5),R5(18),RR5(18),R6(11),RR6(11),PI,XK,AA,XI(5),
     2XR(2),E1,E2,N,NB,NBC,M,MB,NSA
      XN=X*N
      XNB=X*NB
      SR=RSHL(X)
      RC=SR+E1+E2
      IF(NSA)1,1,2
    1 CN= COS(XN)
      CNB= COS(XNB)
      GO TO 3
    2 CN= SIN(XN)
      CNB= SIN(XNB)
    3 CC=CN*CNB
      R3(2)=CC/(RC*RC*SR)
      R3(1)=R3(2)/SR
      RETURN
      END
```

```
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                    +
C     + SUBROUTINE RING4(X)                                                +
C     +                                                                    +
C     + PURPOSE                                                            +
C     +    CALCULATE AT A GIVEN VALUE OF X THE VALUES OF THE INTEGRANDS    +
C     +    OF THE FOURTH SET OF CIRCUMFERENTIAL INTEGRALS OF THE RING      +
C     +                                                                    +
C     + USAGE                                                              +
C     +    IT IS USED AS AN ARGUMENT OF THE NUMERICAL INTEGRATION          +
C     +    SUBROUTINE  'GAUSS'                                             +
C     +                                                                    +
C     + DESCRIPTION OF THE PARAMETERS                                      +
C     +    E1          Z-DISTANCE OF THE SHEAR CENTER OF THE RING FROM THE +
C     +                MIDDLE SURFACE OF THE SHELL                         +
C     +    E2          Z-DISTANCE OF THE CENTROID OF THE RING FROM ITS     +
C     +                SHEAR CENTER                                        +
C     +    X           THE INPUT VALUE OF X AT WHICH THE INTEGRANDS HAVE   +
C     +                TO BE EVALUATED                                     +
C     +    R4(I)       INTEGRANDS                                          +
C     +    N           N                                                   +
C     +    NB          NB                                                  +
C     +    NSA         0 WHEN COMPUTING THE SYMMETRIC  MODE SHAPES         +
C     +                1 WHEN COMPUTING THE ANTISYMMETRIC MODE SHAPES      +
C     +                                                                    +
C     + REMARKS                                                            +
C     +    EXCEPT FOR X ALL THE INPUT AND OUTPUT PARAMETERS ARE            +
C     +    COMMUNICATED THROUGH THE COMMON STATEMENT                       +
C     +                                                                    +
C     + FUNCTION SUBROUTINES REQUIRED                                      +
C     +    RSHL                                                            +
C     +    RSHLT                                                           +
C     +                                                                    +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE RING4(X)
      COMMON DR(9),R(9),DRV(5),RV(5),R1(8),RR1(8),R2(10),RR2(10),R3(2),
     1RR3(2),R4(5),RR4(5),R5(18),RR5(18),R6(11),RR6(11),PI,XK,AA,XI(5),
     2XR(2),E1,E2,N,NB,NBC,M,MB,NSA
      XN=X*N
      XNB=X*NB
      SR=RSHL(X)
      Z=E1+E2
      RC=SR+Z
      SRT=RSHLT(X)
      RCT=-SRT/(RC*RC)
      IF(NSA)1,1,2
    1 CN= COS(XN)
      SN= SIN(XN)
      CNB= COS(XNB)
      SNB= SIN(XNB)
      GO TO 3
    2 CN= SIN(XN)
      SN= COS(XN)
      CNB= SIN(XNB)
      SNB= COS(XNB)
      RCT=-RCT
    3 SS=SN*SNB
      SC=SN*CNB
      CS=CN*SNB
      R4(1)=RCT*RCT*SS/RC
      R4(5)=RCT*CS/RC
      R4(3)=R4(5)/RC
```

```
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                      +
C     + SUBROUTINE RING5(X)                                                  +
C     +                                                                      +
C     + PURPOSE                                                              +
C     +    CALCULATE AT A GIVEN VALUE OF X THE VALUES OF THE INTEGRANDS      +
C     +    OF THE FIFTH  SET OF CIRCUMFERENTIAL INTEGRALS OF THE RING        +
C     +                                                                      +
C     + USAGE                                                                +
C     +    IT IS USED AS AN ARGUMENT OF THE NUMERICAL INTEGRATION            +
C     +    SUBROUTINE  'GAUSS'                                               +
C     +                                                                      +
C     + DESCRIPTION OF THE PARAMETERS                                        +
C     +    E1          Z-DISTANCE OF THE SHEAR CENTER OF THE RING FROM THE   +
C     +                MIDDLE SURFACE OF THE SHELL                           +
C     +    E2          Z-DISTANCE OF THE CENTROID OF THE RING FROM ITS       +
C     +                SHEAR CENTER                                          +
C     +    X           THE INPUT VALUE OF X AT WHICH THE INTEGRANDS HAVE     +
C     +                TO BE EVALUATED                                       +
C     +    R5(I)       INTEGRANDS                                            +
C     +    N           N                                                     +
C     +    NB          NB                                                    +
C     +    NSA         0 WHEN COMPUTING THE SYMMETRIC  MODE SHAPES           +
C     +                1 WHEN COMPUTING THE ANTISYMMETRIC MODE SHAPES        +
C     +                                                                      +
C     + REMARKS                                                              +
C     +    EXCEPT FOR X ALL THE INPUT AND OUTPUT PARAMETERS ARE              +
C     +    COMMUNICATED THROUGH THE COMMON STATEMENT                         +
C     +                                                                      +
C     + FUNCTION SUBROUTINES REQUIRED                                        +
C     +    RSHL                                                              +
C     +    RRRT                                                              +
C     +    RSHLT                                                             +
C     +                                                                      +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE RING5(X)
      COMMON DR(9),R(9),DRV(5),RV(5),R1(8),RR1(8),R2(10),RR2(10),R3(2),
     1RR3(2),R4(5),RR4(5),R5(18),RR5(18),R6(11),RR6(11),PI,XK,AA,XI(5),
     2XR(2),E1,E2,N,NB,NBC,M,MR,NSA
      XN=X*N
      XNB=X*NB
      SR=RSHL(X)
      Z=E1+E2
      RC=SR+Z
      SRT=RSHLT(X)
      RC2=RC*RC
      RCT=-SRT/RC2
      RT=RRRT(X)
      IF(NSA)1,1,2
    1 CN= COS(XN)
      SN= SIN(XN)
      CNB= COS(XNB)
      SNB= SIN(XNB)
      GO TO 3
    2 CN= SIN(XN)
      SN= COS(XN)
      CNB= SIN(XNB)
      SNB= COS(XNB)
      RT=-RT
      RCT=-RCT
    3 SS=SN*SNB
      SC=SN*CNB
```

```
CS=CN*SNB
U=RCT/(RC2*SR)
R5(3)=RCT*RCT*SS/(RC*SR)
R5(2)=R5(3)/SR
R5(18)=SS*RT*RT/RC
R5(1)=R5(18)/RC2
R5(4)=SS*RT*RCT/RC2
R5(5)=R5(4)/SR
R5(6)=SC*RT/RC
R5(7)=CS*RT/RC
R5(11)=R5(7)/SR
R5(10)=R5(6)/SR
R5(8)=R5(6)/RC2
R5(9)=R5(7)/RC2
R5(12)=R5(8)/SR
R5(13)=R5(9)/SR
R5(16)=SC*U
R5(17)=CS*U
R5(14)=R5(16)/SR
R5(15)=R5(17)/SR
RETURN
END
```

```
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                    +
C     + SUBROUTINE RING6(X)                                                +
C     +                                                                    +
C     + PURPOSE                                                            +
C     +    CALCULATE AT A GIVEN VALUE OF X THE VALUES OF THE INTEGRANDS    +
C     +    OF THE SIXTH  SET OF CIRCUMFERENTIAL INTEGRALS OF THE RING      +
C     +                                                                    +
C     + USAGE                                                              +
C     +    IT IS USED AS AN ARGUMENT OF THE NUMERICAL INTEGRATION          +
C     +    SUBROUTINE  'GAUSS'                                             +
C     +                                                                    +
C     + DESCRIPTION OF THE PARAMETERS                                      +
C     +    E1          Z-DISTANCE OF THE SHEAR CENTER OF THE RING FROM THE +
C     +                MIDDLE SURFACE OF THE SHELL                         +
C     +    E2          Z-DISTANCE OF THE CENTROID OF THE RING FROM ITS     +
C     +                SHEAR CENTER                                        +
C     +    X           THE INPUT VALUE OF X AT WHICH THE INTEGRANDS HAVE   +
C     +                TO BE EVALUATED                                     +
C     +    R6(I)       INTEGRANDS                                          +
C     +    N           N                                                   +
C     +    NB          NB                                                  +
C     +    NSA         0 WHEN COMPUTING THE SYMMETRIC  MODE SHAPES         +
C     +                1 WHEN COMPUTING THE ANTISYMMETRIC MODE SHAPES      +
C     +                                                                    +
C     + REMARKS                                                            +
C     +    EXCEPT FOR X ALL THE INPUT AND OUTPUT PARAMETERS ARE            +
C     +    COMMUNICATED THROUGH THE COMMON STATEMENT                       +
C     +                                                                    +
C     + FUNCTION SUBROUTINES REQUIRED                                      +
C     +    RSHL                                                            +
C     +    RRRT                                                            +
C     +    RSHLT                                                           +
C     +                                                                    +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE RING6(X)
      COMMON DR(9),R(9),DRV(5),RV(5),R1(8),RR1(8),R2(10),RR2(10),R3(2),
     1RR3(2),R4(5),RR4(5),R5(18),RR5(18),R6(11),RR6(11),PI,XK,AA,XI(5),
     2XR(2),E1,E2,N,NB,NBC,M,MB,NSA
      XN=X*N
      XNB=X*NB
      SR=RSHL(X)
      Z=E1+E2
      RC=SR+Z
      SRT=RSHLT(X)
      RC2=RC*RC
      RCT=-SRT/RC2
      RT=RRRT(X)
      IF(NSA)1,1,2
    1 CN= COS(XN)
      SN= SIN(XN)
      CNB= COS(XNB)
      SNB= SIN(XNB)
      GO TO 3
    2 CN= SIN(XN)
      SN= COS(XN)
      CNB= SIN(XNB)
      SNB= COS(XNB)
      RT=-RT
      RCT=-RCT
    3 SS=SN*SNB
      SC=SN*CNB
```

```
CS=CN*SNB
U=RCT/(RC*SR)
V=RT/RC2
R6(1)=SS*RT*RT/RC2
R6(2)=SS*RT*RCT/RC
R6(3)=R6(2)/SR
R6(4)=SC*V
R6(5)=CS*V
R6(6)=R6(4)/SR
R6(7)=R6(5)/SR
R6(10)=SC*U
R6(11)=CS*U
R6(8)=R6(10)/SR
R6(9)=R6(11)/SR
RETURN
END
```

```
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     +                                                                      +
C     + SUBROUTINE GAUSS                                                     +
C     +                                                                      +
C     + PURPOSE                                                              +
C     +    TO EVALUATE THE INTEGRALS OF A SET OF FUNCTIONS OVER THE          +
C     +    INTERVAL  LOWL TO UPL                                             +
C     +                                                                      +
C     + USAGE                                                                +
C     +    CALL GAUSS(NG,KG,LOWL,UPL,NOFN,PHI,FX,SUM,ANS,FOFX)               +
C     +    PARAMETER FOFX REQUIRES EXTERNAL STATEMENT                        +
C     +                                                                      +
C     + DESCRIPTION OF THE PARAMETERS                                        +
C     +    NG          NUMBER OF POINTS OF THE GAUSSIAN QUADRATURE           +
C     +    KG          NUMBER OF SUBINTERVALS                                +
C     +    LOWL        LOWER LIMIT OF THE INTEGRATION                        +
C     +    UPL         UPPER LIMIT OF THE INTEGRATION                        +
C     +    NOFN        NUMBER OF FUNCTIONS TO BE INTEGRATED                  +
C     +    PHI         AN INTERMEDIET VALUE                                  +
C     +    FX          NAME OF THE VECTOR WHOSE ELEMENTS ARE THE FUNCTIONS   +
C     +                WHICH ARE TO BE INTEGRATED                            +
C     +    SUM         WORKING VECTOR                                        +
C     +    ANS         THE NAME OF THE OUTPUT VECTOR WHICH CONTAINS THE      +
C     +                VALUES OF THE INTEGRALS                               +
C     +    FOFX        THE NAME OF THE EXTERNAL SUBROUTINE USED TO EXPRESS   +
C     +                THE FUNCTIONS FX                                      +
C     +    TEMP        WORKING VECTOR WHOSE DIMENSION MUST BE GREATER THAN   +
C     +                OR EQUAL TO NOFN                                      +
C     +    CONST       VECTOR CONTAINING THE ABSISSAS AND WEIGHT FACTORS     +
C     +    NLF         WORKING VECTOR                                        +
C     +                                                                      +
C     + REMARKS                                                              +
C     +    THE                                                               +
C     +    THIS SUBROUTINE IS EQUIPPED TO PERFORM 3-,4-,5-,6-,7-,8-,9-,      +
C     +    10-,16-,32-POINT GAUSSIAN QUADRATURE WITH ANY SPECIFIED KG        +
C     +    NUMBER OF SUBINTERVALS                                            +
C     +                                                                      +
C     + SUBROUTINES REQUIRED                                                 +
C     +    THE EXTERNAL SUBROUTINE FOFX CONTAINING A SET OF FUNCTIONS        +
C     +    MUST BE FURNISHED BY THE USER                                     +
C     +                                                                      +
C     + METHOD                                                               +
C     +    THE CONSTANTS (ABSISSAS AND WEIGHT FACTORS) USED IN THIS          +
C     +    SUBROUTINE WERE OBTAINED FROM TABLE 25.4, M. ABRAMOWITZ AND       +
C     +    I. A. STEGUN, HAND BOOK OF MATHEMATICAL FUNCTIONS WITH            +
C     +    FORMULAS, GRAPHS, AND MATHEMATICAL TABLES, U. S. DEPT. OF         +
C     +    COMMERCE, NATIONAL BUREAU OF STANDARDS, APPLIED MATHEMATICS       +
C     +    SERIES. 55.                                                       +
C     +                                                                      +
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE GAUSS  (NG,KG,LOWL,UPL,NOFN,PHI,FX,SUM,ANS,FOFX)
      REAL LOWL
      DIMENSION SUM(1),ANS(1),FX(1),CONST(104),TEMP(35),NLS(32)
C
      DATA NLS /-1,-1,0,4,8,14,20,28,36,46,-1,-1,-1,-1,-1,56,
     *        -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,72 /
      CONST(1)=0.0
      CONST(2)=0.888888888888889
      CONST(3)=0.774596669242483
      CONST(4)=0.555555555555556
      CONST(5)=0.339981043584856
      CONST(6)=0.652145154862546
```

```
CONST(7)=0.861136311594053
CONST(8)=0.347854845137454
CONST(9)=0.0
CONST(10)=0.568888888888889
CONST(11)=0.538469310105683
CONST( 12)=0.478628670499366
CONST( 13)=0.906179845938664
CONST( 14)=0.236926885056189
CONST( 15)=0.238619186083197
CONST( 16)=0.467913934572691
CONST( 17)=0.661209386466265
CONST( 18)=0.360761573048139
CONST( 19)=0.932469514203152
CONST( 20)=0.17132449237917
CONST( 21)=0.000000000000000
CONST( 22)=0.417959183673469
CONST( 23)=0.405845151377397
CONST( 24)=0.381830050505119
CONST( 25)=0.741531185599394
CONST( 26)=0.279705391489277
CONST( 27)=0.949107912342759
CONST( 28)=0.12948496616887
CONST( 29)=0.18343464249565
CONST( 30)=0.362683783378362
CONST( 31)=0.525532409916329
CONST( 32)=0.313706645877887
CONST( 33)=0.796666477413627
CONST( 34)=0.222381034453374
CONST( 35)=0.960289856497536
CONST( 36)=0.101228536290376
CONST( 37)=0.000000000000000
CONST( 38)=0.330239355001260
CONST( 39)=0.324253423403809
CONST( 40)=0.312347077040003
CONST( 41)=0.613371432700590
CONST( 42)=0.260610696402935
CONST( 43)=0.836031107326636
CONST( 44)=0.180648160694857
CONST( 45)=0.968160239507626
CONST( 46)=0.081274388361574
CONST( 47)=0.148874338981631
CONST( 48)=0.295524224714753
CONST( 49)=0.433395394129247
CONST( 50)=0.269266719309996
CONST( 51)=0.679409568299024
CONST( 52)=0.219086362515982
CONST( 53)=0.865063366688985
CONST( 54)=0.149451349150581
CONST( 55)=0.973906528517172
CONST( 56)=0.066671344308688
CONST( 57)=0.095012509837637
CONST( 58)=0.189450610455068
CONST( 59)=0.281603550779259
CONST( 60)=0.182603415044924
CONST( 61)=0.458016777657227
CONST( 62)=0.169156519395003
CONST( 63)=0.617876244402644
CONST( 64)=0.149595988816577
CONST( 65)=0.755404408355003
CONST( 66)=0.124628971255534
CONST( 67)=0.865631202387832
CONST( 68)=0.095158511682493
```

```
      CONST( 69)=0.944575023073233
      CONST( 70)=0.062253523938648
      CONST( 71)=0.989400934991650
      CONST( 72)=0.027152459411754
      CONST( 73)=0.048307665687738
      CONST( 74)=0.096540088514728
      CONST( 75)=0.144471961582796
      CONST( 76)=0.095638720079275
      CONST( 77)=0.239287362252137
      CONST( 78)=0.093844399080805
      CONST( 79)=0.331868602282128
      CONST( 80)=0.091173878695764
      CONST( 81)=0.421351276130635
      CONST( 82)=0.087652093004404
      CONST( 83)=0.506899908932229
      CONST( 84)=0.083311924226947
      CONST( 85)=0.587715757240762
      CONST( 86)=0.078193895787070
      CONST( 87)=0.663044266930215
      CONST( 88)=0.072345794108849
      CONST( 89)=0.732182118740290
      CONST( 90)=0.065822222776362
      CONST( 91)=0.794483795967942
      CONST( 92)=0.058684093478536
      CONST( 93)=0.849367613732570
      CONST( 94)=0.050998059262376
      CONST( 95)=0.896321155766052
      CONST( 96)=0.042835898022227
      CONST( 97)=0.934906075937740
      CONST( 98)=0.034273862913021
      CONST( 99)=0.964762255587506
      CONST(100)=0.025392065309262
      CONST(101)=0.985611511545268
      CONST(102)=0.016274394730906
      CONST(103)=0.997263861849482
      CONST(104)=0.007018610009470
 1001 FORMAT (31H1GAUSSIAN INTEGRATION OF ORDER I5, /
     *         17H0IS NOT AVAILABLE /
     *         17H0PROGRAM STOPPED.)
C
C  FROM ORDER SPECIFIED, FIND START LOCATION IN DATA.
      DO 20 NORD=1,32
      IF (NG .EQ. NORD) GO TO 25
   20 CONTINUE
   99 WRITE (6,1001) NG
      CALL EXIT
   25 LS = ALS(NORD)
      IF (LS .LT. 0) GO TO 99
      FKG= FLOAT(KG)
      DO 1 N=1,NOFN
    1 ANS(N)=0.0
      A1=LOWL
      A2=A1
      B=UPL
      DX=(B-A1)/FKG
      DO 2 I=1,KG
      A1=A2
      B=A1+DX
      A2=B
      ALPHA=0.5E0*DX
      BETA=ALPHA+A1
      DO 3 N=1,NOFN
```

```
 3 SUM(N)=0.0
   DO 4 J=1,NG,2
   LOC = LS + J
   PYKA = CONST(LOC) * ALPHA
   PHI=BETA-PYKA
   CALL FOFX (PHI)
   DO 5 N=1,NOFN
 5 TEMP(N)=FX(N)
   IF(PYKA)7,6,7
 6 DO 8 N=1,NOFN
 8 TEMP(N)=0.0
   GO TO 9
 7 PHI=PYKA+BETA
   CALL FOFX (PHI)
 9 DO 10 N=1,NOFN
   LOCP1 = LS + J+1
10 SUM(N) = SUM(N) + (TEMP(N) + FX(N)) * CONST(LOCP1)
 4 CONTINUE
   DO 11 N=1,NOFN
11 ANS(N)=ANS(N)+SUM(N)*ALPHA
 2 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE EIGENC (AM,AK,MN3,KRRR,VECR,LC,XXXX,Y,MC,Z,EVR,EVI,INDI
     1C)
      DIMENSION AM(KRRR,1),AK(KRRR,1),VECR(KRRR,1),LC(1),EVR(1),XXXX(1),
     1Y(1),MC(1),Z(1),INDIC(1),EVI(1)
C
C
C     THE PURPOSE OF THIS SUBROUTINE IS TO ARRANGE THE MATRICIES INTO
C     THE FORM REQUIRED BY SUBROUTINE EIGENP
C
C
      CALL RRAY(2,MN3,MN3,KRRR,KRRR,XXXX,AM)
      CALL RRAY(2,MN3,MN3,KRRR,KRRR,Y,AK)
      CALL INV(XXXX,MN3,LC,MC)
      CALL MPRD(XXXX,Y,Z,MN3,MN3,MN3)
      CALL RRAY(1,MN3,MN3,KRRR,KRRR,Z,AK)
      DO 2005 IJ=1,MN3
      EVR(IJ)=0.000
      EVI(IJ)=0.000
      DO 2005 KJ=1,MN3
      AM(IJ,KJ)=0.000
 2005 VECR(IJ,KJ)=0.000
      U=48.000
      CALL EIGENP(MN3,KRRR,AK,U,EVR,EVI,VECR,AM,INDIC)
      RETURN
      END
```

```
      SUBROUTINE EIGENP(N,NM,A,T,EVR,EVI,VECR,VECI,INDIC)
C      THIS SUBROUTINE TAKEN FROM COMMUNICATIONS OF ACM VOL. 11 NO. 12,
C      DEC. 68
      INTEGER I,IVEC,J,K,K1,KON,L,L1,M,N,NM
      DIMENSION A(NM,1),VECR(NM,1),VECI(NM,1),
     1EVR(NM),EVI(NM),INDIC(NM)
      DIMENSION IWORK(120),LOCAL(120),PRFACT(120)
     1,SUBDIA(120),WORK1(120),WORK2(120),WORK(120)
C
C THIS SUBROUTINE FINDS ALL THE EIGENVALUES AND THE
C EIGENVECTORS OF A REAL GENERAL MATRIX OF ORDER N.
C
C FIRST IN THE SUBROUTINE SCALE THE MATRIX IS SCALED SO THAT
C THE CORRESPONDING ROWS AND COLUMNS ARE APPROXIMATELY
C BALANCED AND THEN THE MATRIX IS NORMALISED SO THAT THE
C VALUE OF THE EUCLIDIAN NORM OF THE MATRIX IS EQUAL TO ONE.
C
C THE EIGENVALUES ARE COMPUTED BY THE QR DOUBLE-STEP METHOD
C IN THE SUBROUTINE HESQR.
C THE EIGENVECTORS ARE COMPUTED BY INVERSE ITERATION IN
C THE SUBROUTINE REALVE, FOR THE REAL EIGENVALUES,OR IN THE
C SUBROUTINE COMPVE,FOR THE COMPLEX EIGENVALUES.
C
C THE ELEMENTS OF THE MATRIX ARE TO BE STORED IN THE FIRST N
C ROWS AND COLUMNS OF THE TWO DIMENSIONAL ARRAY A. THE
C ORIGINAL MATRIX IS DESTROYED BY THE SUBROUTINE.
C N IS THE ORDER OF THE MATRIX.
C NM DEFINES THE FIRST DIMENSION OF THE TWO DIMENSIONAL
C ARRAYS A,VECR,VECI AND THE DIMENSION OF THE ONE
C DIMENSIONAL ARRAYS EVR,EVI AND INDIC. THEREFORE THE
C CALLING PROGRAM SHOULD CONTAIN THE FOLLOWING DECLARATION
C      DIMENSION A(NM,NN),VECR(NM,NN),VECI(NM,NN),
C     1EVR(NM),EVI(NM),INDIC(NM)
C WHERE NM AND NN ARE ANY NUMBERS EQUAL TO OR GREATER THAN N
C THE UPPER LIMIT FOR NM IS EQUAL TO 100 BUT MAY BE
C INCREASED TO THE VALUE MAX BY REPLACING THE DIMENSION
C STATEMENT
C      DIMENSION IWORK(100),LOCAL(100),....,WORK(100)
C IN THE SUBROUTINE EIGENP WITH
C      DIMENSION IWORK(MAX),LOCAL(MAX),....,WORK(MAX)
C NM AND NN ARE OF COURSE BOUNDED BY THE SIZE OF THE STORE.
C
C THE REAL PARAMETER T MUST BE SET EQUAL TO THE NUMBER OF
C BINARY DIGITS IN THE MANTISSA OF A DOUBLE PRECISION
C FLOATING-POINT NUMBER.
C
C THE REAL PARTS OF THE N COMPUTED EIGENVALUES WILL BE FOUND
C IN THE FIRST N PLACES OF THE ARRAY EVR AND THE IMAGINARY
C PARTS IN THE FIRST N PLACES OF THE ARRAY EVI.
C THE REAL COMPONENTS OF THE NORMALISED EIGENVECTOR I
C (I=1,2,....,N) CORRESPONDING TO THE EIGENVALUE STORED IN
C EVR(I) AND EVI(I) WILL BE FOUND IN THE FIRST N PLACES OF
C THE COLUMN I OF THE TWO DIMENSIONAL ARRAY VECR AND THE
C IMAGINARY COMPONENTS IN THE FIRST N PLACES OF THE COLUMN I
C OF THE TWO DIMENSIONAL ARRAY VECI.
C
C THE REAL EIGENVECTOR IS NORMALISED SO THAT THE SUM OF THE
C SQUARES OF THE COMPONENTS IS EQUAL TO ONE.
C THE COMPLEX EIGENVECTOR IS NORMALISED SO THAT THE
C COMPONENT WITH THE LARGEST VALUE IN MODULUS HAS ITS REAL
C PART EQUAL TO ONE AND THE IMAGINARY PART EQUAL TO ZERO.
C
```

```
C THE ARRAY INDIC INDICATES THE SUCCESS OF THE SUBROUTINE
C EIGENP AS FOLLOWS
C       VALUE OF INDIC(I)      EIGENVALUE I      EIGENVECTOR I
C              0                 NOT FOUND         NOT FOUND
C              1                 FOUND             NOT FOUND
C              2                 FOUND             FOUND
C
C
        IF(N.NE.1)GO TO 1
        EVR(1) = A(1,1)
        EVI(1) = 0.00
        VECR(1,1) = 1.00
        VECI(1,1) = 0.00
        INDIC(1) = 2
        GO TO 25
C
      1 CALL SCALE(N,NM,A,VECI,PRFACT,ENORM)
C THE COMPUTATION OF THE EIGENVALUES OF THE NORMALISED
C MATRIX.
        EX =  EXP(-T*ALOG(2.00))
        CALL HESQR(N,NM,A,VECI,EVR,EVI,SUBDIA,INDIC,EPS,EX)
C
C THE POSSIBLE DECOMPOSITION OF THE UPPER-HESSENBERG MATRIX
C INTO THE SUBMATRICES OF LOWER ORDER IS INDICATED IN THE
C ARRAY LOCAL. THE DECOMPOSITION OCCURS WHEN SOME
C SUBDIAGONAL ELEMENTS ARE IN MODULUS LESS THAN A SMALL
C POSITIVE NUMBER EPS DEFINED IN THE SUBROUTINE HESQR . THE
C AMOUNT OF WORK IN THE EIGENVECTOR PROBLEM MAY BE
C DIMINISHED IN THIS WAY.
        J = N
        I = 1
        LOCAL(1) = 1
        IF(J.EQ.1)GO TO 4
      2 IF( ABS(SUBDIA(J-1)).GT.EPS)GO TO 3
        I = I+1
        LOCAL(I)=0
      3 J = J-1
        LOCAL(I)=LOCAL(I)+1
        IF(J.NE.1)GO TO 2
C
C THE EIGENVECTOR PROBLEM.
      4 K = 1
        KON = 0
        L = LOCAL(1)
        M = N
        DO 10 I=1,N
          IVEC = N-I+1
          IF(I.LE.L)GO TO 5
          K = K+1
          M = N-L
          L = L+LOCAL(K)
      5   IF(INDIC(IVEC).EQ.0)GO TO 10
          IF(EVI(IVEC).NE.0.00)GO TO 8
C
C TRANSFER OF AN UPPER-HESSENBERG MATRIX OF THE ORDER M FROM
C THE ARRAYS VECI AND SUBDIA INTO THE ARRAY A.
          DO 7 K1=1,M
            DO 6 L1=K1,M
      6       A(K1,L1) = VECI(K1,L1)
            IF(K1.EQ.1)GO TO 7
            A(K1,K1-1) = SUBDIA(K1-1)
      7     CONTINUE
```

```
C
C THE COMPUTATION OF THE REAL EIGENVECTOR IVEC OF THE UPPER-
C HESSENBERG MATRIX CORRESPONDING TO THE REAL EIGENVALUE
C EVR(IVEC).
        CALL REALVE(N,NM,M,IVEC,A,VECR,EVR,EVI,IWORK,
     1  WORK,INDIC,EPS,EX)
        GO TO 10
C
C THE COMPUTATION OF THE COMPLEX EIGENVECTOR IVEC OF THE
C UPPER-HESSENBERG MATRIX CORRESPONDING TO THE COMPLEX
C EIGENVALUE EVR(IVEC) + I*EVI(IVEC). IF THE VALUE OF KON IS
C NOT EQUAL TO ZERO THEN THIS COMPLEX EIGENVECTOR HAS
C ALREADY BEEN FOUND FROM ITS CONJUGATE.
    8   IF(KON.NE.0)GO TO 9
        KON = 1
        CALL COMPVE(N,NM,M,IVEC,A,VECR,VECI,EVR,EVI,INDIC,
     1  IWORK,SUBDIA,WORK1,WORK2,WORK,EPS,EX)
        GO TO 10
    9   KON = 0
   10   CONTINUE
C
C THE RECONSTRUCTION OF THE MATRIX USED IN THE REDUCTION OF
C MATRIX A TO AN UPPER-HESSENBERG FORM BY HOUSEHOLDER METHOD
      DO 12 I=1,N
        DO 11 J=I,N
          A(I,J) = 0.00
   11     A(J,I) = 0.00
   12   A(I,I) = 1.00
      IF(N.LE.2)GO TO 15
      M = N-2
      DO 14 K=1,M
        L = K+1
        DO 14 J=2,N
          D1 = 0.00
          DO 13 I=L,N
            D2 = VECI(I,K)
   13       D1 = D1+ D2*A(J,I)
          DO 14 I=L,N
   14       A(J,I) = A(J,I)-VECI(I,K)*D1
C
C THE COMPUTATION OF THE EIGENVECTORS OF THE ORIGINAL NON-
C SCALED MATRIX.
   15 KON = 1
      DO 24 I=1,N
        L = 0
        IF(EVI(I).EQ.0.00)GO TO 16
        L = 1
        IF(KON.EQ.0)GO TO 16
        KON = 0
        GO TO 24
   16   DO 18 J=1,N
          D1 = 0.00
          D2 = 0.00
          DO 17 K=1,N
            D3 = A(J,K)
            D1 = D1+D3*VECR(K,I)
            IF(L.EQ.0)GO TO 17
            D2 = D2+D3*VECR(K,I-1)
   17       CONTINUE
          WORK(J) = D1/PRFACT(J)
          IF(L.EQ.0)GO TO 18
          SUBDIA(J)=D2/PRFACT(J)
```

```
   18      CONTINUE
C
C THE NORMALIZATION OF THE EIGENVECTORS AND THE COMPUTATION
C OF THE EIGENVALUES OF THE ORIGINAL NON-NORMALISED MATRIX.
          IF(L.EQ.1)GO TO 21
          D1 = 0.00
          DO 19 M=1,N
            W1 = WORK(M)
   19       D1 = D1+W1*W1
          D1 =  SQRT(D1)
          DO 20 M=1,N
            VECI(M,I) = 0.00
   20       VECR(M,I) = WORK(M)/D1
          EVR(I) = EVR(I)*ENORM
          GO TO 24
C
   21     KON = 1
          EVR(I) = EVR(I)*ENORM
        EVR(I-1) = EVR(I)
        EVI(I) = EVI(I)*ENORM
        EVI(I-1) =-EVI(I)
        R = 0.00
        DO 22 J=1,N
          W1 = WORK(J)
          W2 = SUBDIA(J)
          R1 = W1*W1 + W2*W2
          IF(R.GE.R1)GO TO 22
          R = R1
          L = J
   22     CONTINUE
        D3 = WORK(L)
        R1 = SUBDIA(L)
        DO 23 J=1,N
          D1 = WORK(J)
          D2 = SUBDIA(J)
          VECR(J,I) = (D1*D3+D2*R1)/R
          VECI(J,I) = (D2*D3-D1*R1)/R
          VECR(J,I-1) = VECR(J,I)
   23   VECI(J,I-1) =-VECI(J,I)
   24     CONTINUE
C
   25   RETURN
        END
```

```
      SUBROUTINE SCALE(N,NM,A,H,PRFACT,ENORM)
      INTEGER I,J,ITER,N,NCOUNT,NM
      DIMENSION A(NM,1),H(NM,1),PRFACT(NM)
C
C THIS SUBROUTINE STORES THE MATRIX OF THE ORDER N FROM THE
C ARRAY A INTO THE ARRAY H. AFTERWARD THE MATRIX IN THE
C ARRAY A IS SCALED SO THAT THE QUOTIENT OF THE ABSOLUTE SUM
C OF THE OFF-DIAGONAL ELEMENTS OF COLUMN I AND THE ABSOLUTE
C SUM OF THE OFF-DIAGONAL ELEMENTS OF ROW I LIES WITHIN THE
C VALUES OF BOUND1 AND BOUND2.
C THE COMPONENT I OF THE EIGENVECTOR OBTAINED BY USING THE
C SCALED MATRIX MUST BE DIVIDED BY THE VALUE FOUND IN THE
C PRFACT(I) OF THE ARRAY PRFACT. IN THIS WAY THE EIGENVECTOR
C OF THE NON-SCALED MATRIX IS OBTAINED.
C
C AFTER THE MATRIX IS SCALED IT IS NORMALISED SO THAT THE
C VALUE OF THE EUCLIDIAN NORM IS EQUAL TO ONE.
C IF THE PROCESS OF SCALING WAS NOT SUCCESSFUL THE ORIGINAL
C MATRIX FROM THE ARRAY H WOULD BE STORED BACK INTO A AND
C THE EIGENPROBLEM WOULD BE SOLVED BY USING THIS MATRIX.
C NM DEFINES THE FIRST DIMENSION OF THE ARRAYS A AND H. NM
C MUST BE GREATER OR EQUAL TO N.
C THE EIGENVALUES OF THE NORMALISED MATRIX MUST BE
C MULTIPLIED BY THE SCALAR ENORM IN ORDER THAT THEY BECOME
C THE EIGENVALUES OF THE NON-NORMALISED MATRIX.
C
      DO 2 I=1,N
        DO 1 J=1,N
    1     H(I,J) =A(I,J)
    2   PRFACT(I)= 1.00
      BOUND1 = 0.7500
      BOUND2 = 1.3300
      ITER = 0
    3 NCOUNT = 0
      DO 9 I=1,N
        COLUMN = 0.00
        ROW = 0.00
        DO 4 J=1,N
          IF(I.EQ.J)GO TO 4
          COLUMN = COLUMN +   ABS(A(J,I))
          ROW = ROW +   ABS(A(I,J))
    4     CONTINUE
        IF(COLUMN.EQ.0.00)GO TO 5
          IF(ROW.EQ.0.00)GO TO 5
          Q = COLUMN/ROW
          IF(Q.LT.BOUND1)GO TO 6
          IF(Q.GT.BOUND2)GO TO 6
    5   NCOUNT = NCOUNT + 1
        GO TO 8
    6   FACTOR =   SQRT(Q)
        DO 7 J=1,N
          IF(I.EQ.J)GO TO 7
          A(I,J) = A(I,J)*FACTOR
          A(J,I) = A(J,I)/FACTOR
    7     CONTINUE
        PRFACT(I) = PRFACT(I)*FACTOR
    8   CONTINUE
      ITER = ITER+1
      IF(ITER.GT.30)GO TO 11
      IF(NCOUNT.LT.N)GO TO 3
C
      ENORM = 0.00
```

```
      DO 9 I=1,N
        DO 9 J=1,N
          Q=A(I,J)
    9     FNORM = FNORM+Q*Q
      FNORM =  SQRT(FNORM)
      DO 10 I=1,N
        DO 10 J=1,N
   10     A(I,J)=A(I,J)/FNORM
      FNORM = FNORM
      GO TO 13
C
   11 DO 12 I=1,N
        PRFACT(I)=1.00
        DO 12 J=1,N
   12     A(I,J) = H(I,J)
      FNORM = 1.00
C
   13 RETURN
      END
```

```
      SUBROUTINE HESQR(N,NM,A,H,EVR,EVI,SUBDIA,INDIC,EPS,EX)
      INTEGER I,J,K,L,M,MAXST,M1,N,NM,NS
      DIMENSION A(NM,1),H(NM,1),EVR(NM),EVI(NM),SUBDIA(NM)
      DIMENSION INDIC(NM)
C
C THIS SUBROUTINE FINDS ALL THE EIGENVALUES OF A REAL
C GENERAL MATRIX. THE ORIGINAL MATRIX A OF ORDER N IS
C REDUCED TO THE UPPER-HESSENBERG FORM H BY MEANS OF
C SIMILARITY TRANSFORMATIONS(HOUSEHOLDER METHOD). THE MATRIX
C H IS PRESERVED IN THE UPPER HALF OF THE ARRAY H AND IN THE
C ARRAY SUBDIA. THE SPECIAL VECTORS USED IN THE DEFINITION
C OF THE HOUSEHOLDER TRANSFORMATION MATRICES ARE STORED IN
C THE LOWER PART OF THE ARRAY H.
C NM IS THE FIRST DIMENSION OF THE ARRAYS A AND H. NM MUST
C BE EQUAL TO OR GREATER THAN N.
C THE REAL PARTS OF THE N EIGENVALUES WILL BE FOUND IN THE
C FIRST N PLACES OF THE ARRAY EVR, AND
C THE IMAGINARY PARTS IN THE FIRST N PLACES OF THE ARRAY EVI.
C THE ARRAY INDIC INDICATES THE SUCCESS OF THE ROUTINE AS
C FOLLOWS
C      VALUE OF INDIC(I)      EIGENVALUE I
C             0                 NOT FOUND
C             1                   FOUND
C EPS IS A SMALL POSITIVE NUMBER THAT NUMERICALLY REPRESENTS
C ZERO IN THE PROGRAM. EPS = (EUCLIDIAN NORM OF H)*EX ,WHERE
C EX = 2**(-T). T IS THE NUMBER OF BINARY DIGITS IN THE
C MANTISSA OF A FLOATING POINT NUMBER.
C
C
C
C REDUCTION OF THE MATRIX A TO AN UPPER-HESSENBERG FORM H.
C THERE ARE N-2 STEPS.
      IF(N-2)14,1,2
    1 SUBDIA(1) = A(2,1)
      GO TO 14
    2 M = N-2
      DO 12 K=1,M
        L = K+1
        S = 0.00
        DO 3 I=L,N
          H(I,K) = A(I,K)
    3     S = S+ ABS(A(I,K))
        IF(S.NE. ABS(A(K+1,K)))GO TO 4
        SUBDIA(K) = A(K+1,K)
        H(K+1,K) = 0.00
        GO TO 12
    4   SR2 = 0.00
        DO 5 I=L,N
          SR = A(I,K)
          SR = SR/S
          A(I,K) = SR
    5     SR2 = SR2+SR*SR
        SR =   SQRT(SR2)
        IF(A(L,K).LT.0.00)GO TO 6
        SR = -SR
    6   SR2 = SR2-SR*A(L,K)
        A(L,K) = A(L,K)-SR
        H(L,K) = H(L,K)-SR*S
        SUBDIA(K) = SR*S
        X = S* SQRT(SR2)
        DO 7 I=L,N
          H(I,K) = H(I,K)/X
```

```
    7      SUBDIA(I) = A(I,K)/SR2
C PREMULTIPLICATION BY THE MATRIX PR.
        DO 9 J=L,N
           SR = 0.00
           DO 8 I=L,N
    8        SR = SR+A(I,K)*A(I,J)
           DO 9 I=L,N
    9        A(I,J) = A(I,J)-SUBDIA(I)*SR
C POSTMULTIPLICATION BY THE MATRIX PR.
        DO 11 J=1,N
           SR=0.00
           DO 10 I=L,N
   10        SR = SR+A(J,I)*A(I,K)
           DO 11 I=L,N
   11        A(J,I) = A(J,I)-SUBDIA(I)*SR
   12   CONTINUE
      DO 13 K=1,M
   13   A(K+1,K) = SUBDIA(K)
C TRANSFER OF THE UPPER HALF OF THE MATRIX A INTO THE
C ARRAY H AND THE CALCULATION OF THE SMALL POSITIVE NUMBER
C EPS.
      SUBDIA(N-1) = A(N,N-1)
   14 EPS = 0.00
      DO 15 K=1,N
        INDIC(K) = 0
        IF(K.NE.N)EPS = EPS + SUBDIA(K)*SUBDIA(K)
        DO 15 I=K,N
          H(K,I) = A(K,I)
          W2 = A(K,I)
   15     EPS = EPS + W2*W2
      EPS = EX* SQRT(EPS)
C
C THE QR ITERATIVE PROCESS. THE UPPER-HESSENBERG MATRIX H IS
C REDUCED TO THE UPPER-MODIFIED TRIANGULAR FORM.
C
C DETERMINATION OF THE SHIFT OF ORIGIN FOR THE FIRST STEP OF
C THE QR ITERATIVE PROCESS.
      SHIFT = A(N,N-1)
      IF(N.LE.2)SHIFT = 0.00
      IF(A(N,N).NE.0.00)SHIFT = 0.00
      IF(A(N-1,N).NE.0.00)SHIFT = 0.00
      IF(A(N-1,N-1).NE.0.00)SHIFT = 0.00
      M = N
      NS= 0
      MAXST = N*10
C
C TESTING IF THE UPPER HALF OF THE MATRIX IS EQUAL TO ZERO.
C IF IT IS EQUAL TO ZERO THE QR PROCESS IS NOT NECESSARY.
      DO 16 I=2,N
        DO 16 K=I,N
          IF(A(I-1,K).NE.0.00)GO TO 18
   16     CONTINUE
      DO 17 I=1,N
        INDIC(I)=1
        EVR(I) = A(I,I)
   17   EVI(I) = 0.00
      GO TO 37
C
C START THE MAIN LOOP OF THE QR PROCESS.
   18 K=M-1
      M1=K
      I = K
```

```
C FIND ANY DECOMPOSITIONS OF THE MATRIX.
C JUMP TO 34 IF THE LAST SUBMATRIX OF THE DECOMPOSITION IS
C OF THE ORDER ONE.
C JUMP TO 35 IF THE LAST SUBMATRIX OF THE DECOMPOSITION IS
C OF THE ORDER TWO.
      IF(K)37,34,19
   19 IF( ABS(A(M,K)).LE.EPS)GO TO 34
      IF(M-2.EQ.0)GO TO 35
   20   I = I-1
        IF( ABS(A(K,I)).LE.EPS)GO TO 21
        K = I
        IF(K.GT.1)GO TO 20
   21 IF(K.EQ.M1)GO TO 35
C TRANSFORMATION OF THE MATRIX OF THE ORDER GREATER THAN TWO
      S = A(M,M)+A(M1,M1)+SHIFT
      SR = A(M,M)*A(M1,M1)-A(M,M1)*A(M1,M) + 0.2500*SHIFT*SHIFT
      A(K+2,K) = 0.00
C CALCULATE X1,Y1,Z1,FOR THE SUBMATRIX OBTAINED BY THE
C DECOMPOSITION.
      X = A(K,K)*(A(K,K)-S)+A(K,K+1)*A(K+1,K)+SR
      Y = A(K+1,K)*(A(K,K)+A(K+1,K+1)-S)
      R =  ABS(X)+ ABS(Y)
      IF(R.EQ.0.00)SHIFT = A(M,M-1)
      IF(R.EQ.0.00)GO TO 21
      Z = A(K+2,K+1)*A(K+1,K)
      SHIFT = 0.00
      NS =NS+1
C
C THE LOOP FOR ONE STEP OF THE QR PROCESS.
      DO 33 I=K,M1
        IF(I.EQ.K)GO TO 22
C CALCULATE XR,YR,ZR.
        X = A(I,I-1)
        Y = A(I+1,I-1)
        Z = 0.00
        IF(I+2.GT.M)GO TO 22
        Z = A(I+2,I-1)
   22   SR2 =  ABS(X)+ ABS(Y)+ ABS(Z)
        IF(SR2.EQ.0.00)GO TO 23
        X = X/SR2
        Y = Y/SR2
        Z = Z/SR2
   23   S =  SQRT(X*X + Y*Y +Z*Z)
        IF(X.LT.0.00)GO TO 24
        S = -S
   24   IF(I.EQ.K)GO TO 25
        A(I,I-1) = S*SR2
   25   IF (SR2.NE.0.00)GO TO 26
        IF(I+3.GT.M)GO TO 33
        GO TO 32
   26   SR = 1.00-X/S
        S = X-S
        X = Y/S
        Y = Z/S
C PREMULTIPLICATION BY THE MATRIX PR.
        DO 28 J=I,M
          S = A(I,J)+A(I+1,J)*X
          IF(I+2.GT.M)GO TO 27
          S = S+A(I+2,J)*Y
   27     S = S*SR
          A(I,J) = A(I,J)-S
          A(I+1,J) = A(I+1,J)-S*X
```

```
           IF(I+2.GT.M)GC TO 28
           A(I+2,J) = A(I+2,J)-S*Y
   28      CCNTINUE
C POSTMULTIPLICATION BY THE MATRIX PR.
         L = I+2
         IF(I.LT.M1)GO TO 29
         L = M
   29    DO 31 J=K,L
           S = A(J,I)+A(J,I+1)*X
           IF(I+2.GT.M)GO TO 3C
           S = S + A(J,I+2)*Y
   30      S = S*SR
           A(J,I) = A(J,I)-S
           A(J,I+1)=A(J,I+1)-S*X
           IF(I+2.GT.M)GC TO 31
           A(J,I+2)=A(J,I+2)-S*Y
   31      CONTINUE
         IF(I+3.GT.M)GO TO 33
         S = -A(I+3,I+2)*Y*SR
   32    A(I+3,I) = S
         A(I+3,I+1) = S*X
         A(I+3,I+2) = S*Y + A(I+3,I+2)
   33    CONTINUF
C
       IF(NS.GT.MAXST)GO TO 37
       GO TO 18
C
C COMPUTE THE LAST EIGENVALUE.
   34 EVR(M) = A(M,M)
       EVI(M) = 0.00
       INDIC(M) = 1
       M = K
       GO TO 18
C
C CCMPUTE THE EIGENVALUES CF THE LAST 2X2 MATRIX OBTAINED BY
C THE DECOMPOSITION.
   35 R = 0.500*(A(K,K)+A(M,M))
       S = 0.500*(A(M,M)-A(K,K))
       S = S*S + A(K,M)*A(M,K)
       INDIC(K) = 1
       INDIC(M) = 1
       IF(S.LT.0.00)GO TO 36
       T =  SQRT(S)
       EVR(K) = R-T
       EVR(M) = R+T
       EVI(K) = 0.00
       EVI(M) = 0.00
       M = M-2
       GC TC 18
   36 T =  SQRT(-S)
       EVR(K) = R
       EVI(K) = T
       EVR(M) = R
       EVI(M) = -T
       M = M-2
       GC TC 18
C
   37 RETURN
       END
```

```
      SUBROUTINE REALVE(N,NM,M,IVEC,A,VECR,EVR,EVI,
     1IWORK,WORK,INDIC,EPS,EX)
      INTEGER I,IVEC,ITER,J,K,L,M,N,NM,NS
      DIMENSION A(NM,1),VECR(NM,1),EVR(NM)
      DIMENSION EVI(NM),IWORK(NM),WORK(NM),INDIC(NM)
C
C THIS SUBROUTINE FINDS THE REAL EIGENVECTOR OF THE REAL
C UPPER-HESSENBERG MATRIX IN THE ARRAY A,CORRESPONDING TO
C THE REAL EIGENVALUE STORED IN EVR(IVEC). THE INVERSE
C ITERATION METHOD IS USED.
C NOTE THE MATRIX IN A IS DESTROYED BY THE SUBROUTINE.
C N IS THE ORDER OF THE UPPER-HESSENBERG MATRIX.
C NM DEFINES THE FIRST DIMENSION OF THE TWO DIMENSIONAL
C ARRAYS A AND VECR. NM MUST BE EQUAL TO OR GREATER THAN N.
C M IS THE ORDER OF THE SUBMATRIX OBTAINED BY A SUITABLE
C DECOMPOSITION OF THE UPPER-HESSENBERG MATRIX IF SOME
C SUBDIAGONAL ELEMENTS ARE EQUAL TO ZERO. THE VALUE OF M IS
C CHOSEN SO THAT THE LAST N-M COMPONENTS OF THE EIGENVECTOR
C ARE ZERO.
C IVEC GIVES THE POSITION OF THE EIGENVALUE IN THE ARRAY EVR
C FOR WHICH THE CORRESPONDING EIGENVECTOR IS COMPUTED.
C THE ARRAY EVI WOULD CONTAIN THE IMAGINARY PARTS OF THE N
C EIGENVALUES IF THEY EXISTED.
C
C THE M COMPONENTS OF THE COMPUTED REAL EIGENVECTOR WILL BE
C FOUND IN THE FIRST M PLACES OF THE COLUMN IVEC OF THE TWO
C DIMENSIONAL ARRAY VECR.
C
C IWORK AND WORK ARE THE WORKING STORES USED DURING THE
C GAUSSIAN ELIMINATION AND BACKSUBSTITUTION PROCESS.
C THE ARRAY INDIC INDICATES THE SUCCESS OF THE ROUTINE AS
C FOLLOWS
C      VALUE OF INDIC(I)       EIGENVECTOR I
C             1                  NOT FOUND
C             2                  FOUND
C EPS IS A SMALL POSITIVE NUMBER THAT NUMERICALLY REPRESENTS
C ZERO IN THE PROGRAM. EPS = (EUCLIDIAN NORM OF A)*EX,WHERE
C EX = 2**(-T). T IS THE NUMBER OF BINARY DIGITS IN THE
C      MANTISSA OF A FLOATING POINT NUMBER.
      VECR(1,IVEC) = 1.D0
      IF(M.EQ.1)GO TO 24
C SMALL PERTURBATION OF EQUAL EIGENVALUES TO OBTAIN A FULL
C SET OF EIGENVECTORS.
      EVALUE = EVR(IVEC)
      IF(IVEC.EQ.M)GO TO 2
      K = IVEC+1
      R = 0.D0
      DO 1 I=K,M
        IF(EVALUE.NE.EVR(I))GO TO 1
        IF(EVI(I).NE.0.D0)GO TO 1
        R = R+3.D0
    1   CONTINUE
      EVALUE = EVALUE+R*EX
    2 DO 3 K=1,M
    3   A(K,K) = A(K,K)-EVALUE
C
C GAUSSIAN ELIMINATION OF THE UPPER-HESSENBERG MATRIX A. ALL
C ROW INTERCHANGES ARE INDICATED IN THE ARRAY IWORK.ALL THE
C MULTIPLIERS ARE STORED AS THE SUBDIAGONAL ELEMENTS OF A.
      K = M-1
      DO 8 I=1,K
        L = I+1
```

```
          IWORK(I) = 0
          IF(A(I+1,I).NE.0.00)GO TO 4
          IF(A(I,I).NE.0.00)GO TO 8
          A(I,I) = EPS
          GO TO 8
    4     IF( ABS(A(I,I)).GE. ABS(A(I+1,I)))GO TO 6
          IWORK(I) = 1
          DO 5 J=I,M
            R = A(I,J)
            A(I,J) = A(I+1,J)
    5       A(I+1,J) = R
    6     R = -A(I+1,I)/A(I,I)
          A(I+1,I) = R
          DO 7 J=L,M
    7       A(I+1,J) = A(I+1,J)+R*A(I,J)
    8     CONTINUE
        IF(A(M,M).NE.0.00)GO TO 9
        A(M,M) = EPS
C
C THE VECTOR (1,1,...,1) IS STORED IN THE PLACE OF THE RIGHT
C HAND SIDE COLUMN VECTOR.
    9 DO 11 I=1,N
          IF(I.GT.M)GO TO 10
          WORK(I) = 1.00
          GO TO 11
   10     WORK(I) = 0.00
   11     CONTINUE
C
C THE INVERSE ITERATION IS PERFORMED ON THE MATRIX UNTIL THE
C INFINITE NORM OF THE RIGHT-HAND SIDE VECTOR IS GREATER
C THAN THE BOUND DEFINED AS  0.01/(N*EX).
        BOUND = 0.0100/(EX *  FLOAT(N))
        NS = 0
        ITER = 1
C
C THE BACKSUBSTITUTION.
   12 R = 0.00
        DO 15 I=1,M
          J = M-I+1
          S = WORK(J)
          IF(J.EQ.M)GO TO 14
          L = J+1
          DO 13 K=L,M
            SR = WORK(K)
   13       S  = S - SR*A(J,K)
   14     WORK(J) = S/A(J,J)
          T =  ABS(WORK(J))
          IF(R.GE.T)GO TO 15
          R = T
   15     CONTINUE
C
C THE COMPUTATION OF THE RIGHT-HAND SIDE VECTOR FOR THE NEW
C ITERATION STEP.
          DO 16 I=1,M
   16     WORK(I) = WORK(I)/R
C
C THE COMPUTATION OF THE RESIDUALS AND COMPARISON OF THE
C RESIDUALS OF THE TWO SUCCESSIVE STEPS OF THE INVERSE
C ITERATION.IF THE INFINITE NORM OF THE RESIDUAL VECTOR IS
C GREATER THAN THE INFINITE NORM OF THE PREVIOUS RESIDUAL
C VECTOR THE COMPUTED EIGENVECTOR OF THE PREVIOUS STEP IS
C TAKEN AS THE FINAL EIGENVECTOR.
```

```
      R1 = 0.00
      CO 18 I=1,M
        T = 0.00
        DO 17 J=I,M
 17       T = T+A(I,J)*WORK(J)
        T =  ABS(T)
        IF(R1.GE.T)GO TO 18
        R1= T
 18     CONTINUE
      IF(ITER.EQ.1)GO TO 19
      IF(PREVIS.LE.R1)GO TO 24
 19   DO 20 I=1,M
 20     VECR(I,IVEC) = WORK(I)
      PREVIS = R1
      IF(NS.EQ.1)GO TO 24
      IF(ITER.GT.6)GO TO 25
      ITER = ITER+1
      IF(R.LT.BOUND)GO TO 21
      NS = 1
C
C GAUSSIAN ELIMINATION OF THE RIGHT-HAND SIDE VECTOR.
 21   K = M-1
      DO 23 I=1,K
        R = WORK(I+1)
      IF(IWORK(I).EQ.0)GO TO 22
        WORK(I+1)=WORK(I)+WORK(I+1)*A(I+1,I)
        WORK(I) = R
        GO TO 23
 22     WORK(I+1)=WORK(I+1)+WORK(I)*A(I+1,I)
 23     CONTINUE
      GO TO 12
C
 24   INDIC(IVEC) = 2
 25   IF(M.EQ.N)GO TO 27
      J = M+1
      DO 26 I=J,N
 26     VECR(I,IVEC) = 0.00
 27   RETURN
      END
```

```
      SUBROUTINE COMPVE(N,NM,M,IVEC,A,VECR,H,EVR,EVI,INDIC,
     1IWORK,SUBDIA,WORK1,WORK2,WORK,EPS,EX)
      INTEGER I,I1,I2,ITER,IVEC,J,K,L,M,N,NM,NS
      DIMENSION A(NM,1),VECR(NM,1),H(NM,1),EVR(NM),EVI(NM),
     1INDIC(NM),IWORK(NM),SUBDIA(NM),WORK1(NM),WORK2(NM),
     2WORK(NM)
C
C THIS SUBROUTINE FINDS THE COMPLEX EIGENVECTOR OF THE REAL
C UPPER-HESSENBERG MATRIX OF ORDER N CORRESPONDING TO THE
C COMPLEX EIGENVALUE WITH THE REAL PART IN EVR(IVEC) AND THE
C CORRESPONDING IMAGINARY PART IN EVI(IVEC). THE INVERSE
C ITERATION METHOD IS USED MODIFIED TO AVOID THE USE OF
C COMPLEX ARITHMETIC.
C THE MATRIX ON WHICH THE INVERSE ITERATION IS PERFORMED IS
C BUILT UP IN THE ARRAY A BY USING THE UPPER-HESSENBERG
C MATRIX PRESERVED IN THE UPPER HALF OF THE ARRAY H AND IN
C THE ARRAY SUBDIA.
C NM DEFINES THE FIRST DIMENSION OF THE TWO DIMENSIONAL
C ARRAYS A,VECR AND H. NM MUST BE EQUAL TO OR GREATER
C THAN N.
C M IS THE ORDER OF THE SUBMATRIX OBTAINED BY A SUITABLE
C DECOMPOSITION OF THE UPPER-HESSENBERG MATRIX IF SOME
C SUBDIAGONAL ELEMENTS ARE EQUAL TO ZERO. THE VALUE OF M IS
C CHOSEN SO THAT THE LAST N-M COMPONENTS OF THE COMPLEX
C EIGENVECTOR ARE ZERO.
C
C THE REAL PARTS OF THE FIRST M COMPONENTS OF THE COMPUTED
C COMPLEX EIGENVECTOR WILL BE FOUND IN THE FIRST M PLACES OF
C THE COLUMN WHOSE TOP ELEMENT IS VECR(1,IVEC) AND THE
C CORRESPONDING IMAGINARY PARTS OF THE FIRST M COMPONENTS OF
C THE COMPLEX EIGENVECTOR WILL BE FOUND IN THE FIRST M
C PLACES OF THE COLUMN WHOSE TOP ELEMENT IS VECR(1,IVEC-1).
C
C THE ARRAY INDIC INDICATES THE SUCCESS OF THE ROUTINE AS
C FOLLOWS
C      VALUE OF INDIC(I)      EIGENVECTOR I
C             1                  NOT FOUND
C             2                    FOUND
C THE ARRAYS IWORK,WORK1,WORK2 AND WORK ARE THE WORKING
C STORES USED DURING THE INVERSE ITERATION PROCESS.
C EPS IS A SMALL POSITIVE NUMBER THAT NUMERICALLY REPRESENTS
C ZERO IN THE PROGRAM. EPS = (EUCLIDIAN NORM OF H)*EX, WHERE
C EX = 2**(-T). T IS THE NUMBER OF BINARY DIGITS IN THE
C MANITISSA OF A FLOATING POINT NUMBER.
C
      FKSI = EVR(IVEC)
      ETA  = EVI(IVEC)
C THE MODIFICATION OF THE EIGENVALUE (FKSI + I*ETA) IF MORE
C EIGENVALUES ARE EQUAL.
      IF(IVEC.EQ.M)GO TO 2
      K = IVEC+1
      R = 0.00
      DO 1 I=K,M
        IF(FKSI.NE.EVR(I))GO TO 1
        IF( ABS(ETA).NE. ABS(EVI(I)))GO TO 1
        R = R + 3.00
    1   CONTINUE
      R = R*EX
      FKSI = FKSI+R
      ETA  = ETA +R
C
C THE MATRIX  ((H-FKSI*I)*(H-FKSI*I) + (ETA*ETA)*I)  IS
```

```
C STORED INTO THE ARRAY A.
    2 R = FKSI*FKSI + ETA*ETA
      S = 2.0D*FKSI
      L = M-1
      DO 5 I=1,M
        DO 4 J=I,M
          D = 0.D0
          A(J,I) = 0.D0
          DO 3 K = I,J
    3       D = D+H(I,K)*H(K,J)
    4     A(I,J) = D-S*H(I,J)
    5   A(I,I) = A(I,I)+R
      DO 9 I=1,L
        R = SUBDIA(I)
        A(I+1,I) = -S*R
        I1 = I+1
        DO 6 J=1,I1
    6     A(J,I) = A(J,I)+R*H(J,I+1)
        IF(I.EQ.1)GO TO 7
        A(I+1,I-1) = R*SUBDIA(I-1)
    7   DO 8 J=I,M
    8     A(I+1,J) = A(I+1,J)+R*H(I,J)
    9   CONTINUE
C
C THE GAUSSIAN ELIMINATION OF THE MATRIX
C ((H-FKSI*I)*(H-FKSI*I) + (ETA*ETA)*I) IN THE ARRAY A. THE
C ROW INTERCHANGES THAT OCCUR ARE INDICATED IN THE ARRAY
C IWORK. ALL THE MULTIPLIERS ARE STORED IN THE FIRST AND IN
C THE SECOND SUBDIAGONAL OF THE ARRAY A.
      K = M-1
      DO 18 I=1,K
        I1 = I+1
        I2 = I+2
        IWORK(I) = 0
        IF(I.EQ.K)GO TO 10
        IF(A(I+2,I).NE.0.D0)GO TO 11
   10   IF(A(I+1,I).NE.0.D0)GO TO 11
        IF(A(I,I).NE.0.D0)GO TO 18
        A(I,I) = EPS
        GO TO 18
C
   11   IF(I.EQ.K)GO TO 12
        IF( ABS(A(I+1,I)).GE. ABS(A(I+2,I)))GO TO 12
        IF( ABS(A(I,I)).GE. ABS(A(I+2,I)))GO TO 16
        L = I+2
        IWORK(I) = 2
        GO TO 13
   12   IF( ABS(A(I,I)).GE. ABS(A(I+1,I)))GO TO 15
        L = I+1
        IWORK(I) = 1
C
   13   DO 14 J=I,M
          R = A(I,J)
          A(I,J) = A(L,J)
   14     A(L,J) = R
   15   IF(I.NE.K)GO TO 16
        I2 = I1
   16   DO 17 L=I1,I2
          R = -A(L,I)/A(I,I)
          A(L,I) = R
          DO 17 J=I1,M
   17       A(L,J) = A(L,J)+R*A(I,J)
```

```
   18   CONTINUE
        IF(A(M,M).NE.0.00)GO TO 19
        A(M,M) = EPS
C
C THE VECTOR (1,1,...,1) IS STORED INTO THE RIGHT-HAND SIDE
C VECTORS VECR( ,IVEC) AND VECR( ,IVEC-1) REPRESENTING THE
C COMPLEX RIGHT-HAND SIDE VECTOR.
   19 DO 21 I=1,N
        IF(I.GT.M)GO TO 20
        VECR(I,IVEC) = 1.00
        VECR(I,IVEC-1) = 1.00
        GO TO 21
   20   VECR(I,IVEC) = 0.00
        VECR(I,IVEC-1) = 0.00
   21   CONTINUE
C
C THE INVERSE ITERATION IS PERFORMED ON THE MATRIX UNTIL THE
C INFINITE NORM OF THE RIGHT-HAND SIDE VECTOR IS GREATER
C THAN THE BOUND DEFINED AS 0.01/(N*EX).
      BOUND = 0.0100/(EX* FLOAT(N))
      NS = 0
      ITER = 1
      DO 22 I=1,M
   22   WORK(I) = H(I,I)-FKSI
C
C THE SEQUENCE OF THE COMPLEX VECTORS Z(S) = P(S)+I*Q(S) AND
C W(S+1)= U(S+1)+I*V(S+1) IS GIVEN BY THE RELATIONS
C (A - (FKSI-I*ETA)*I)*W(S+1) = Z(S)  AND
C Z(S+1) = W(S+1)/MAX(W(S+1)).
C THE FINAL W(S) IS TAKEN AS THE COMPUTED EIGENVECTOR.
C
C THE COMPUTATION OF THE RIGHT-HAND SIDE VECTOR
C (A-FKSI*I)*P(S)-ETA*Q(S). A IS AN UPPER-HESSENBERG MATRIX.
   23 DO 27 I=1,M
        D = WORK(I)*VECR(I,IVEC)
        IF(I.EQ.1)GO TO 24
        D = D+SUBDIA(I-1)*VECR(I-1,IVEC)
   24   L = I+1
        IF(L.GT.M)GO TO 26
        DO 25 K=L,M
   25     D = D+H(I,K)*VECR(K,IVEC)
   26   VECR(I,IVEC-1) = D-ETA*VECR(I,IVEC-1)
   27   CONTINUE
C
C GAUSSIAN ELIMINATION OF THE RIGHT-HAND SIDE VECTOR.
      K = M-1
      DO 28 I=1,K
        L = I+IWORK(I)
        R = VECR(L,IVEC-1)
        VECR(L,IVEC-1) = VECR(I,IVEC-1)
        VECR(I,IVEC-1) = R
        VECR(I+1,IVEC-1) = VECR(I+1,IVEC-1)+A(I+1,I)*R
        IF(I.EQ.K)GO TO 28
        VECR(I+2,IVEC-1) = VECR(I+2,IVEC-1)+A(I+2,I)*R
   28   CONTINUE
C
C THE COMPUTATION OF THE REAL PART U(S+1) OF THE COMPLEX
C VECTOR W(S+1). THE VECTOR U(S+1) IS OBTAINED AFTER THE
C BACKSUBSTITUTION.
      DO 31 I=1,M
        J = M-I+1
        D = VECR(J,IVEC-1)
```

```
      IF(J.EQ.M)GO TO 30
      L = J+1
      DO 29 K=L,M
        D1 = A(J,K)
29      D = D-D1*VECR(K,IVEC-1)
30    VECR(J,IVEC-1) = D/A(J,J)
31    CONTINUE
C
C THE COMPUTATION OF THE IMAGINARY PART V(S+1) OF THE VECTOR
C W(S+1),WHERE  V(S+1) = (P(S)-(A-FKSI*I)*U(S+1))/ETA.
      DO 35 I=1,M
        D = WORK(I)*VECR(I,IVEC-1)
        IF(I.EQ.1)GO TO 32
        D = D+SUBDIA(I-1)*VECR(I-1,IVEC-1)
32      L = I+1
        IF(L.GT.M)GO TO 34
        DO 33 K=L,M
33        D = D+H(I,K)*VECR(K,IVEC-1)
34      VECR(I,IVEC) = (VECR(I,IVEC)-D)/ETA
35      CONTINUE
C
C THE COMPUTATION OF (INFIN. NORM OF W(S+1))**2 .
      L = 1
      S = 0.00
      DO 36 I=1,M
        W1 = VECR(I,IVEC)
        W2 = VECR(I,IVEC-1)
        R = W1*W1 + W2*W2
        IF(R.LE.S)GO TO 36
        S = R
        L = I
36      CONTINUE
C THE COMPUTATION OF THE VECTOR Z(S+1),WHERE Z(S+1)= W(S+1)/
C (COMPONENT OF W(S+1) WITH THE LARGEST ABSOLUTE VALUE) .
      U = VECR(L,IVEC-1)
      V = VECR(L,IVEC)
      DO 37 I=1,M
        B = VECR(I,IVEC)
        R = VECR(I,IVEC-1)
        VECR(I,IVEC) = (R*U + B*V)/S
37      VECR(I,IVEC-1) = (B*U-R*V)/S
C THE COMPUTATION OF THE RESIDUALS AND COMPARISON OF THE
C RESIDUALS OF THE TWO SUCCESSIVE STEPS OF THE INVERSE
C ITERATION. IF THE INFINITE NORM OF THE RESIDUAL VECTOR IS
C GREATER THAN THE INFINITE NORM OF THE PREVIOUS RESIDUAL
C VECTOR THE COMPUTED VECTOR OF THE PREVIOUS STEP IS TAKEN
C AS THE COMPUTED APPROXIMATION TO THE EIGENVECTOR.
      B = 0.00
      DO 41 I=1,M
        R = WORK(I)*VECR(I,IVEC-1) - ETA*VECR(I,IVEC)
        U = WORK(I)*VECR(I,IVEC) + ETA*VECR(I,IVEC-1)
        IF(I.EQ.1)GO TO 38
        R = R+SUBDIA(I-1)*VECR(I-1,IVEC-1)
        U = U+SUBDIA(I-1)*VECR(I-1,IVEC)
38      L = I+1
        IF(L.GT.M)GO TO 40
        DO 39 J=L,M
          R = R+H(I,J)*VECR(J,IVEC-1)
39        U = U+H(I,J)*VECR(J,IVEC)
40      U = R*R + U*U
        IF(B.GE.U)GO TO 41
        B = U
```

```
41    CONTINUE
      IF(ITER.EQ.1)GO TO 42
      IF(PREVIS.LE.B)GO TO 44
   42 DO 43 I=1,N
        WORK1(I) = VECR(I,IVEC)
   43   WORK2(I) = VECR(I,IVEC-1)
      PREVIS = B
      IF(NS.EQ.1)GO TO 46
      IF(ITER.GT.6)GO TO 47
      ITER = ITER+1
      IF(BOUND.GT. SQRT(S))GO TO 23
      NS = 1
      GO TO 23
C
   44 DO 45 I=1,N
        VECR(I,IVEC) = WORK1(I)
   45   VECR(I,IVEC-1)=WORK2(I)
   46 INDIC(IVEC-1) = 2
      INDIC(IVEC)     = 2
   47 RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE RRAY
C
C        PURPOSE
C           CONVERT DATA ARRAY FROM SINGLE TO DOUBLE DIMENSION OR VICE
C           VERSA.  THIS SUBROUTINE IS USED TO LINK THE USER PROGRAM
C           WHICH HAS DOUBLE DIMENSION ARRAYS AND THE SSP SUBROUTINES
C           WHICH OPERATE ON ARRAYS OF DATA IN A VECTOR FASHION.
C
C        USAGE
C           CALL ARRAY (MODE,I,J,N,M,S,D)
C
C        DESCRIPTION OF PARAMETERS
C           MODE - CODE INDICATING TYPE OF CONVERSION
C                    1 - FROM SINGLE TO DOUBLE DIMENSION
C                    2 - FROM DOUBLE TO SINGLE DIMENSION
C           I    - NUMBER OF ROWS IN ACTUAL DATA MATRIX
C           J    - NUMBER OF COLUMNS IN ACTUAL DATA MATRIX
C           N    - NUMBER OF ROWS SPECIFIED FOR THE MATRIX D IN
C                  DIMENSION STATEMENT
C           M    - NUMBER OF COLUMNS SPECIFIED FOR THE MATRIX D IN
C                  DIMENSION STATEMENT
C           S    - IF MODE=1, THIS VECTOR IS INPUT WHICH CONTAINS THE
C                  ELEMENTS OF A DATA MATRIX OF SIZE I BY J. COLUMN I+1
C                  OF DATA MATRIX FOLLOWS COLUMN I, ETC. IF MODE=2,
C                  THIS VECTOR IS OUTPUT REPRESENTING A DATA MATRIX OF
C                  SIZE I BY J CONTAINING ITS COLUMNS CONSECUTIVELY.
C                  THE LENGTH OF S IS IJ, WHERE IJ=I*J.
C           D    - IF MODE=1, THIS MATRIX OF SIZE N BY M IS OUTPUT,
C                  CONTAINING A DATA MATRIX OF SIZE I BY J IN THE FIRST
C                  I ROWS AND J COLUMNS. IF MODE=2, THIS N BY M MATRIX
C                  IS INPUT CONTAINING A DATA MATRIX OF SIZE I BY J IN
C                  THE FIRST I ROWS AND J COLUMNS.
C
C        REMARKS
C           VECTOR S CAN BE IN THE SAME LOCATION AS MATRIX D.  VECTOR S
C           IS REFERRED AS A MATRIX IN OTHER SSP ROUTINES, SINCE IT
C           CONTAINS A DATA MATRIX.
C           THIS SUBROUTINE CONVERTS ONLY GENERAL DATA MATRICES (STORAGE
C           MODE OF 0).
C
C        SUBROUTINES AND FUNCTION SUBROUTINES REQUIRED
C           NONE
C
C        METHOD
C           REFER TO THE DISCUSSION ON VARIABLE DATA SIZE IN THE SECTION
C           DESCRIBING OVERALL RULES FOR USAGE IN THIS MANUAL.
C
C     ..................................................................
C
      SUBROUTINE RRAY (MODE,I,J,N,M,S,D)
      DIMENSION S(1),D(1)
C
      NI=N-I
C
C        TEST TYPE OF CONVERSION
C
      IF(MODE-1) 100, 100, 120
C
C        CONVERT FROM SINGLE TO DOUBLE DIMENSION
```

```
C
  100 IJ=I*J+1
      NM=N*J+1
      DO 110 K=1,J
      NM=NM-NI
      DO 110 L=1,I
      IJ=IJ-1
      NM=NM-1
  110 D(NM)=S(IJ)
      GO TO 140
C
C         CONVERT FROM DOUBLE TO SINGLE DIMENSION
C
  120 IJ=0
      NM=0
      DO 130 K=1,J
      DO 125 L=1,I
      IJ=IJ+1
      NM=NM+1
  125 S(IJ)=D(NM)
  130 NM=NM+NI
C
  140 RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE MPRD
C
C        PURPOSE
C           MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GENERAL
C           MATRIX
C
C        USAGE
C           CALL MPRD(A,B,R,N,M,L)
C
C        DESCRIPTION OF PARAMETERS
C           A - NAME OF FIRST INPUT MATRIX
C           B - NAME OF SECOND INPUT MATRIX
C           R - NAME OF OUTPUT MATRIX
C           N - NUMBER OF ROWS IN A
C           M - NUMBER OF COLUMNS IN A AND ROWS IN B
C           L - NUMBER OF COLUMNS IN B
C
C        REMARKS
C           ALL MATRICES MUST BE STORED AS GENERAL MATRICES
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B
C           NUMBER OF COLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF ROW
C           OF MATRIX B
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE M BY L MATRIX B IS PREMULTIPLIED BY THE N BY M MATRIX A
C           AND THE RESULT IS STORED IN THE N BY L MATRIX R.
C
C     ..................................................................
C
      SUBROUTINE MPRD (A,B,R,N,M,L)
      DIMENSION A(1),B(1),R(1)
C
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=0
      DO 10 I=1,M
      JI=JI+N
      IB=IB+1
   10 R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END
```

```
C
C     ..................................................................
C
C        SLBROLTINE INV
C
C        PURPCSE
C           INVFRT A MATRIX
C
C        USAGE
C           CALL INV (A,N,L,M)
C
C        DESCRIPTION OF PARAMETERS
C           A - INPUT MATRIX, DESTROYED IN CCMPUTATION AND REPLACED BY
C               RESULTANT INVERSE.
C           N - ORDER CF MATRIX A
C           D - RESULTANT DETERMINANT
C           L - WORK VECTOR OF LENGTH N
C           M - WORK VECTCR OF LENGTH N
C
C        REMARKS
C           MATRIX A MUST BE A GENERAL MATRIX
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NCNE
C
C        METHOD
C           THE STANDARD GAUSS-JORDAN METHOC IS USED. THE DETERMINANT
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
C           THE MATRIX IS SINGULAR.
C
C     ..................................................................
C
      SUBROUTINE INV (A,N,L,M)
      DIMENSION A(1),L(1),M(1)
C
C        ...............................................................
C
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C        STATEMENT WHICH FOLLOWS.
C
C     DOUBLE PRECISION A,D,BIGA,HOLD
C
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C        ROUTINE.
C
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT
C        10 MUST BE CHANGED TO DABS.
C
C        ...............................................................
C
C        SEARCH FOR LARGEST ELEMENT
C
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
```

```
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C        CONTAINED IN BIGA)
C
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
C
C        REDUCE MATRIX
C
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
C
C        DIVIDE ROW BY PIVOT
```

```
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C
C        PRODUCT OF PIVOTS
C
      D=D*BIGA
C
C        REPLACE PIVOT BY RECIPROCAL
C
      A(KK)=1.0/BIGA
   80 CONTINUE
C
C        FINAL ROW AND COLUMN INTERCHANGE
C
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END
```

```
      FUNCTION RSHL(T)
      A = 14.39
      B = 9.35
      EPS=1.0-(B/A)*(B/A)
      RSHL=B*B/(A*(1.0-EPS* COS(T)* COS(T))**1.5)
      RETURN
      END


      FUNCTION RRRT(T)
      A = 14.39
      B = 9.35
      EPS=1.0-(B/A)*(B/A)
      RRRT= SQRT(1.0-EPS* COS(T)* COS(T))*A*EPS* SIN(2.0*T)*1.5/(B*B)
      RETURN
      END


      FUNCTION RSHLT(T)
      A = 14.39
      B = 9.35
      EPS=1.0-(B/A)*(B/A)
      RSHLT=-1.5*B*B*EPS* SIN(2.0*T)/(A*(1.0-EPS* COS(T)* COS(T))**2.5)
      RETURN
      END
```

APPENDIX C

DICTIONARY OF VARIABLES USED IN THE MAIN PROGRAM

AA	Length of the shell
ABN	$n\bar{n}$
ABNA	$n^2\bar{n}$
ABNB	$n\bar{n}^2$
ABN2	$n^2\bar{n}^2$
AK(MN3,MN3)	Stiffness matrix, dimension should be = KRRR
AM(MN3,MN3)	Mass matrix, dimension should be = KRRR
AN	n
AN2	n^2
AR(K)	Cross-sectional area of the k^{th} kind or ring, dimension should be ≥ NK
AS(L)	Cross-sectional area of the ℓ^{th} kind of stringer, dimension should be ≥ NL
BC(2,4)	Data block defining various boundary conditions, dimension should be 2 × 4
BCR(2)	The name of the boundary condition read-in, dimension should be 2
BN	$\bar{n}$
BN2	$\bar{n}^2$
C(8)	Temporary work vector used in the stringer equations, dimension should be 8

CC	CN × CNB
CG	Temporary work variable
CN	Cos (TN) for NSA = 0; Sin (TN) for NSA = 1
CNB	Cos (TNB) for NSA = 0; Sin (THB) for NSA = 1
CR(K,40)	C_1 to C_{40}, constants used in the ring equations (C3), and are defined in Appendix D of Reference 1. The first dimension should be ≥ NK, and the second should be 40.
CS	CN × SNB
D	Isotropic plate flexural stiffness
DR(9)	Vector of the integrands of the circumferential integrals $IS1_1$ to $IS1_9$, dimension should be 9
DRV(5)	Vector of the integrands of the circumferential integrals $IS2_1$ to $IS2_5$, dimension should be 5
E1R(K)	z-distance of the shear center of the k^{th} kind of ring from the middle surface of the shell, dimension should be ≥ NK
E2R(K)	z-distance of the centroid of the k^{th} kind of ring from its shear center, dimension should be ≥ NK
E1RK	Z_{1rk}
E2RK	Z_{2rk}
EC	Shell Young's Modulus
ER(K)	Young's modulus of k^{th} kind of ring, dimension should be ≥ NK
ES(L)	Young's modulus of ℓ^{th} kind of stringer, dimension should be ≥ NL
EVI(MN3)	Vector of imaginary part of the eigenvalues, dimension should be = KRRR

EVR(MN3)	Vector of real part of the eigenvalues, dimension should be = KRRR
GJR(K)	The torsional stiffness of the k^{th} kind of ring, dimension should be ≥ NK
GJS(L)	The torsional stiffness of the ℓ^{th} kind of stringer, dimension should be ≥ NL
H	Shell thickness
I	Row index of [A], [D], [E], [N], [NN], and [P] submatrices
IBC	Temporary work variable
IM	Temporary work variable
IN	Row index of [B], [F], [Q], and [R] submatrices
INDIC(MN3)	This array indicates the success of the subroutine EIGENP as follows:

INDIC(I)	EIGENVALUE I	EIGNEVECTOR I
0	not found	not found
1	found	not found
2	found	found

INN	Row index of [C] and [S] submatrices
IR	(see the listing of the program)
ITEMP, IY1, IY2, IZ1, IZ2	Temporary work variables
J	Column index of [A], and [N] submatrices
JBC	Temporary work variable
JN	Column index of [D], [B], [NN], and [Q] submatrices
JNN	Column index of [E], [F], [C], [P], [R], and [S] submatrices

JTEMP	Temporary work variable
KG,KK	(see the listing of the program)
KQ	Temporary work variable
KRRR	Dimension of [AK], [AM], [VECR], {EVR}, {EVI}, {INDIC}, {LC}, and {MC} $\geq$ MN3
LC(MN3)	Temporary vector used by INV (matrix inversion) subroutine, dimension should be = KRRR
LL	(See listing of the program)
MC(MN3)	Temporary vector used by INV (matric inversion) subroutine, dimension should be = KRRR
MD	Temporary work variable
MMAX, MMIN	(See the listing of the program)
MN3	Order of the mass, stiffness, and modal matrices
MS	Total number of axial mode components considered in the displacement series
MSA	(See the listing of the program)
NBC	The code number assigned for different boundary conditions as follows: 1 for clamped-free 2 for freely supported 3 for clamped-clamped 4 for free-free
NCHNG, ND NDC,NEC	Temporary work variables
NEO	(See the listing of the program)
NEIXT	Temporary work variable

NG, NK, NL NMAX, NMIN	(See the listing of the program)
NNK (K)	Vector of the number of rings of k^{th} kind of ring, dimension should be > NK
NNL(L)	Vector of the number of stringers of ℓ^{th} kind of stringer, dimension should be $\geq$ NL
NNR(I)	Temporary vector containing centroidal information of different kinds of rings, dimension should be $\geq$ NK
NQUIT	1 in the 80th column of a blank card, which when placed at the end of sets of data, signifies the end of the data
NR(NK,NK)	Temporary vector containing centroidal information of the rings, dimension should be NK $\times$ NK
NS	Total number of circumferential mode components considered in the displacement series
NSA,NWEV, NWK, NWM	(See the listing of the program)
PC	Mass density of shell
PHI	Temporary work variable
PI	π = 3.14159
PI2	2π
PR(K)	Mass density of k^{th} kind of ring, dimension should be $\geq$ NK
PS(L)	Mass density of ℓ^{th} kind of stringer, dimension should be $\geq$ NL
R(9)	Vector of the circumferential integrals $IS1_1$ to $IS1_9$, dimension should be = 9

R1(8), R2(10), R3(2), R4(5), R5(18), R6(11)	The vectors of the integrands of the circumferential integrals of the ring, $IR1_1$-$IR1_8$, $IR2_1$-$IR2_{10}$, $IR3_1$ and $IR3_2$, $IR4_1$-$IR4_5$, $IR5_1$-$IR5_{18}$, and $IR6_1$-$IR6_{11}$, respectively; dimensions should be = 8, 10, 2, 5, 18, and 11, respectively
RCG(K)	Vector of centroidal distances of various kinds of rings, dimension should be ≥ NK
RI(K,54)	Temporary work vector for saving the 54 ring integrals $IR1_1$ to $IR6_{11}$. The first dimension should be ≥ NK, and the second dimension should be = 54
RING1 to RING6	Subroutines defining the integrands of the circumferential integrals of the ring, $IR1_1$ to $IR6_{11}$
RR1(8), RR2(10) RR3(2), RR4(5) RR5(18), and RR6(11)	The vectors of the circumferential integrals of the ring, $IR1_1$ - $IR1_8$, $IR2_1$ - $IR2_{10}$, $IR3_1$ and $IR3_2$, $IR4_1$ - $IR4_5$, $IR5_1$ - $IR5_{18}$, and $IR6_1$ - $IR6_{11}$, respectively; dimensions should be equal to 8, 10, 2, 5, 18 and 11, respectively.
RRRT (θ)	Function subroutine furnished by the user of the program to evaluate $(1/R)_{,\theta}$ at a given value of θ
RSHL(θ)	Function subroutine furnished by the user of the program to evaluate R at a given value of θ
RV(5)	Vector of circumferential integrals of the shell, $IS2_1$ to $IS2_5$, dimension should be 5
RX(K,I)	Array of the x-locations of the k^{th} kind of rings, the first dimension should be ≥ NK, and the second dimensions should be ≥ the largest element of the vector NNK(K)

S1 to S8	S_1 to S_8, constants used in Eqs. (C1), and are defined in Appendix D of Volume I
SC	SN × CNB
SHELL1, SHELL2	Subroutines defining the integrands of the circumferential integrals $IS1_1$ to $IS2_5$ of the shell equations
SN	Sin (TN) for NSA = 0; Cos (TN) for NSA = 1
SNB	Sin (TNB) for NSA = 0; Cos (TNB) for NSA = 1
SR	Radius of the shell, R
SR2	R^2
SS(1,30)	SS_1 to SS_{30}, constants used in Eqs. (C2), and are defined in Appendix D of Volume I. The first dimension should be ≥ NL, and the second dimension should be = 30
SSS	SN × SNB
ST(75)	Intermediate terms of the stringer equations, dimension should be = 75
SUM(18)	Temporary work vector used by GAUSS (numerical integration) subroutine, dimension should be = 18
T(L,I)	Array of the θ-locations of the ℓ^{th} kind of stringers, the first dimension should be ≥ NL, and the second dimension should be ≥ the largest element of the vector NNL(L)
TITLE1(7)	Title of the run
TITLE2(7)	Title of the run continued
TN	n × T (L,I)
TNB	$-\bar{n}$ × T (L,I)

TS(L,42)	T_1 to T_{42}, constants used in Eqs. (C2) and are defined in Appendix D of Volume I, the first dimension should be ≥ NL, and the second dimension should be = 42
U	Equal to number of binary digits in the mantissa of a double precision, floating point number
VECR(MN3, MN3)	Eigenvector (modal) matrix, dimension should be = KRRR × KRRR
X(5,IM)	A temporary work matrix, containing the longitudinal integrals IX_1 to IX_5 for every combination of m and $\bar{m}$; the first dimension should be = 5, and the second dimension should be = MS × (MS + 1)/2
X1 to X5	IX_1 to IX_5 longitudinal integrals
XIR(K)	The moment of the k^{th} kind of ring cross-sectional area about an axis parallel to X-axis passing through its centroid
XK	X-location of the k^{th} ring
XNU	Poisson's ratio
XR(1), XR(2)	Temporary work vector used for transferring X_1 and X_2 values from XX subroutine to the main program
XX1, XX2	X_1, X_2 (see eqs. (C8) in Appendix C of Volume I)
XXX(2, K, IM)	A temporary storage three dimensional matrix containing the quantities X_1 and X_2 for every combination of m and $\bar{m}$; the first dimension should be = 2, the second be ≥ NK, and the third should be $= \frac{(MS + 1)\ MS}{2}$
XXXX($MN3^2$)	Temporary work vector, dimension should be = ($MN3^2$)

$Y(MN3^2)$ Temporary work vector, dimension should be = $(MN3^2)$

Y1S(L) y-distance of the shear center of the ℓ^{th} kind of stringer from the z-axis passing through its point of attachment, dimension should be ≥ NL

Y2S(L) y-distance of the centroid of the ℓ^{th} kind of stringer from the shear center, dimension should be ≥ NL

YIS(L) The moment of inertia of the ℓ^{th} kind of stringer cross-sectional area about an axis parallel to y-axis passing through its centroid, dimension should be ≥ NL

YZIS(L) Product of inertia of the ℓ^{th} kind of stringer cross-sectional area about y and z axes passing through its centroid, dimension should be ≥ NL

$Z(MN3^2)$ Temporary work vector, dimension should be = $(MN3)^2$

Z1S(L) z-distance of the shear center of the ℓ^{th} kind of stringer from the middle surface of the shell

Z2S(L) z-distance of the centroid of the ℓ^{th} kind of stringer from its shear center

ZERO 0.0, lower limit of the circumferential integrals of shell and ring

ZIR(K) The moment of inertia of the k^{th} kind of ring cross-sectional area about z or z' axes, dimension should be ≥ NK

ZIS(L) The moment of inertia of the ℓ^{th} kind of stringer cross-sectional area about an axis parallel to z-axis passing through its centroid, dimension should be ≥ NL

APPENDIX D

PREPARATION OF DATA FOR THE PROGRAM OF THE FREE VIBRATIONS OF RING- AND/OR STRINGER STIFFENED NONCIRCULAR CYLINDERS WITH ARBITRARY END CONDITIONS

	DATA	No. of CARDS	FORMAT	ITEMS ON DATA CARD
GENERAL	(a) Name of the boundary condition	1	2A10, 59x, I1	BCR, NQUIT
	(b) General input parameters	1	20I4	NG, KG, LL, NL, KK, NK, NMIN, MMAX, MSA, NMIN, NMAX, NSA, NEW, IR, NWK, NWM, NWEV
	(c) Title of the run	2	7A10/ 7A10	TITLE 1, TITLE 2
SHELL	(a) Geometric and material properties of shell	1	5E15.8	PC, EC, XNU, H, AA
STRINGER	(a) Number of ℓ^{th} kind of stringers	1	I4	NNL(L)
	(b) List of θ-locations of ℓ^{th} kind of stringers	$\frac{NNL(L)}{5}$	5E15.8	(T(L,I),I = 1, NNL(L))

	DATA	No. of CARDS	FORMAT	ITEMS ON DATA CARD
STRINGER	(c) Geometric and material properties of ℓ^{th} kind of stringer	3	5E15.8	PS(L), ES(L), AS(L), Z1S(L), Z2S(L), Y1S(L), Y2S(L), ZIS(L), YIS(L), YZIS(L), GJS(L)
RING	(a) Number of k^{th} kind of rings	1	I4	NNK(K)
	(b) List of X-locations of k^{th} kind of rings	$\frac{NNK(K)}{5}$	5E15.8	(RX(K,I), I = 1, NNK(K))
	(c) Geometric and material properties of k^{th} kind of ring	2	5E15.8	PR(K), ER(K), AR(K), EIR(K), E2R(K), ZIR(K), XIR(K), GJR(K)

Note: (1) All numerical data must be right justified.

(2) No blank spaces are left before and in between data fields.

APPENDIX E

COMPUTER OUTPUT

```
***********************************************************************************

FREE VIBRATIONAL ANALYSIS OF STIFFENED OR UNSTIFFENED CIRCULAR OR
      NONCIRCULAR CYLINDERS WITH ARBITRARY END CONDITIONS

***********************************************************************************
```

GENERAL INPUT INFORMATION

NG =	8	KG =	4	LL =	16	NL =	1	KK =	11
NK =	1	MMIN =	1	MMAX =	9	MSA =	1	NMIN =	1
NMAX =	11	NSA =	0	NEU =	1	IR =	1	NWK =	0
NWM =	0	NWEV =	0						

S H E L L   D A T A

SEWALL'S 16 STRINGER AND 11 RING STIFFENED ELLIPTICAL

CYLINDER WITH A = 14.39   B = 9.35

MASS DENSITY	=	0.25880000D-03 LB SEC.**2/IN.**4
MODULUS OF ELASTICITY	=	0.10000000D 08 LB/IN.**2
POISSON'S RATIO	=	0.30000000D 00
THICKNESS	=	0.32000000D-01 INCHES
LENGTH	=	0.24000000D 02 INCHES
END CONDITIONS	=	FREELY SUPPORTED

```
                    S T R I N G E R   D A T A
                    -------------------------

             (THE UNITS ARE SAME AS THOSE OF SHELL DATA)

                  TOTAL NUMBER OF STRINGERS =    16
         NUMBER OF DIFFERENT KINDS OF STRINGERS =     1
================================================================

            16 STRINGERS WITH THE FOLLOWING PROPERTIES

MASS DENSITY       =  0.25880000D-03  MOD. OF ELAS.   =  0.10600000D 08
AREA               =  0.10368714D 00  SHEAR CTR. (Z1)= -0.47500000D-01
SHEAR CTR. (Y1) =  0.0                CENTROID (Z2)   = -0.23398959D 00
CENTROID   (Y2)  =  0.0               INERTIA (IZZ)   =  0.12850717D-02
INERTIA   (IYY)  =  0.59571042D-02  PROD.INER.(IYZ)=  0.0
               TORSIONAL STIFFNESS =  0.91250000D 03

LOCATED AT FOLLOWING THETA VALUES (DEGREES)

0.0              0.13500000D 02  0.25500000D 02  0.51800000D 02
0.90000000D 02   0.12820000D 03  0.15450000D 03  0.16650000D 03
0.18000000D 03   0.19350000D 03  0.20550000D 03  0.23180000D 03
0.27000000D 03   0.30820000D 03  0.33450000D 03  0.34650000D 03

================================================================
```

RING DATA

(THE UNITS ARE SAME AS THOSE OF SHELL DATA)

TOTAL NUMBER OF RINGS = 11
NUMBER OF DIFFERENT KINDS OF RINGS = 1

==

11 RINGS WITH THE FOLLOWING PROPERTIES

MASS DENSITY	= 0.25880000D-03	MOD. OF ELAST. =	0.10600000D 08
AREA	= 0.10368714D 00	SHEAR CTR. (E1) =	-0.47500000D-01
CENTROID (E2)	= -0.23398959D 00	INERTIA (IZZ) =	0.12850717D-02
INERTIA (IXX)	= 0.59571042D-02	TORS. STIF.(GJ) =	0.91250000D 03

LOCATED AT FOLLOWING X VALUES (INCHES)

0.20000000D 01	0.40000000D 01	0.60000000D 01	0.80000000D 01
0.10000000D 02	0.12000000D 02	0.14000000D 02	0.16000000D 02
0.18000000D 02	0.20000000D 02	0.22000000D 02	

==

EIGENVALUES IN HERTZ

0.52520199D 05	0.46001645D 05	0.48647189D 05	0.43615754D 05
0.44380886D 05	0.38765373D 05	0.33284153D 05	0.34088304D 05
0.32525308D 05	0.30957820D 05	0.31355072D 05	0.31116436D 05
0.30791375D 05	0.29178184D 05	0.29855700D 05	0.26700237D 05
0.27063028D 05	0.25944952D 05	0.25079986D 05	0.23984945D 05
0.22754725D 05	0.23370204D 05	0.22005879D 05	0.22316741D 05
0.19719529D 05	0.19871019D 05	0.19280332D 05	0.18814871D 05
0.18529353D 05	0.18360459D 05	0.17885671D 05	0.17250155D 05
0.17705185D 05	0.16391201D 05	0.16783938D 05	0.15473856D 05
0.15781160D 05	0.15254811D 05	0.15181264D 05	0.14755735D 05
0.14446294D 05	0.14544683D 05	0.14275007D 05	0.14159022D 05
0.13243886D 05	0.13636494D 05	0.12730499D 05	0.13097659D 05
0.12062902D 05	0.11074605D 05	0.10750339D 05	0.10828769D 05
0.10310473D 05	0.99187302D 04	0.97074202D 04	0.98689557D 04
0.81109218D 04	0.79299923D 04	0.78144635D 04	0.74933650D 04
0.75843528D 04	0.61886842D 04	0.65935045D 04	0.60086731D 04
0.60282354D 04	0.55418262D 04	0.49742984D 04	0.51537126D 04
0.47115724D 04	0.44711073D 04	0.40752417D 04	0.43794579D 03
0.74099874D 03	0.97427626D 03	0.11548679D 04	0.13398175D 04
0.17028734D 04	0.17390697D 04	0.20087592D 04	0.37004882D 04
0.24136185D 04	0.36158813D 04	0.34718714D 04	0.33142711D 04
0.28413854D 04	0.28562459D 04	0.29236699D 04	0.31133282D 04
0.29588204D 04	0.31082799D 04		

```
     00000000011111111112222222222333333333344444444445555555555666666666677777777778
     12345678901234567890123456789012345678901234567890123456789012345678901234567890
CARD
0001 FREELY SUPPORTED
0002     8   4  16   1  11   1   1   9   1   1  11   0   1   1   0   0   0
0003       SEWALL'S 16 STRINGER AND 11 RING STIFFENED ELLIPTICAL
0004                  CYLINDER WITH A = 14.39  B = 9.35
0005  0.25880000D-03 0.10000000D+08 0.30000000D+00 0.03200000D+00 0.24000000D+02
0006   16
0007  0.00000000D+00 0.13500000D+02 0.25500000D+02 0.51800000D+02 0.90000000D+02
0008  0.12820000D+03 0.15450000D+03 0.16650000D+03 0.18000000D+03 0.19350000D+03
0009  0.20550000D+03 0.23180000D+03 0.27000000D+03 0.30820000D+03 0.33450000D+03
0010  0.34650000D+03
0011  0.25880000D-03 0.10600000D+08 0.10368714D+00-0.04750000D+00-0.23398959D+00
0012  0.00000000D+00 0.00000000D+00 0.12850717D-02 0.59571042D-02 0.00000000D+00
0013  0.91250000D+03
0014   11
0015  0.20000000D+01 0.40000000D+01 0.60000000D+01 0.80000000D+01 0.10000000D+02
0016  0.12000000D+02 0.14000000D+02 0.16000000D+02 0.18000000D+02 0.20000000D+02
0017  0.22000000D+02
0018  0.25880000D-03 0.10600000D+08 0.10368714D+00-0.04750000D+00-0.23398959D+00
0019  0.12850717D-02 0.59571042D-02 0.91250000D+03
0020                                                                              1
```